BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

KOREAN PATENT ABSTRACTS (KR)

Document Code:A
(11) Publication No.1020000011728 (43) Publication.Date. 20000225
(21) Application No.1019990028615 (22) Application Date. 19990715
(51) IPC Code:
G06T 7/20

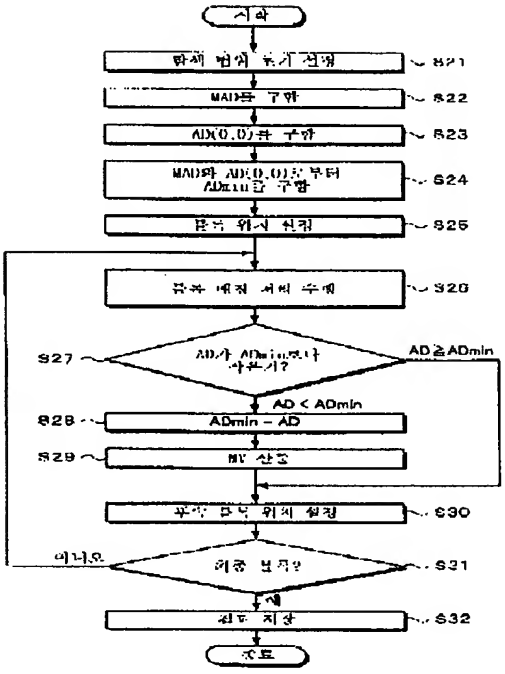
(71) Applicant:
SONY CORPORATION
(72) Inventor:
ANDO YUJI
YASDA HIROYUKI

(30) Priority:
98 200460 19980715 JP
98 212378 19980728 JP
98 223397 19980806 JP
98 223398 19980806 JP
98 235551 19980821 JP
98 241482 19980827 JP

(54) Title of Invention

MOVING VECTOR DETECTING METHOD, RECORD MEDIUM STORING A MOVING VECTOR DETECTING PROGRAM, MOVING DETECTING DEVICE, MOVING DETECTING METHOD, PICTURE ENCODING DEVICE, PICTURE ENCODING METHOD, MOVING VECTOR CALCULATING METHOD, AND RECORD MEDIUM STORING A MOVING VECTOR CALCULATING PROGRAM

Representative drawing



(57) Abstract:

PURPOSE: A moving vector calculating method is provided to be suitable to perform an encoding process according to a moving picture experts group 2(MPEG2) by use of a soft program.

CONSTITUTION: The moving vector calculating method comprises the steps of: extracting a block a reference picture corresponding to a block of a present processed picture; obtaining the remaining between the block of the present picture and the block of the reference picture while moving the block of the reference picture in a predetermined searching region; detecting a block having a minimum remaining from the reference picture so as to calculate a moving vector; converting pixel data of the block of the reference picture and pixel data of the block of the present picture; and obtaining a remaining between data in the block of the reference picture thus converted and data in the block of the present picture thus converted.

if display of image is failed, press (F5)

(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(51) Int. Cl. [6]
G06T 7 /20

(11) 공개번호 특2000-0011728
(43) 공개일자 2000년02월25일

(21) 출원번호 10-1999-0028615
(22) 출원일자 1999년07월15일

(30) 우선권주장 1998-200460 1998년07월15일 일본(JP)
1998-212378 1998년07월28일 일본(JP)
1998-223397 1998년08월06일 일본(JP)
1998-223398 1998년08월06일 일본(JP)
1998-235551 1998년08월21일 일본(JP)
1998-241482 1998년08월27일 일본(JP)

(71) 출원인 소니 가부시끼 가이샤 이데이 노부유끼
일본국 도쿄도 시나가와쿠 키타시나가와 6쵸메 7반 35고

(72) 발명자 안도유지
일본도꾜도시나가와꾸기따시나가와6쪼메7-35소니가부시끼가이샤내
야스다히로유끼
일본도꾜도시나가와꾸기따시나가와6쪼메7-35소니가부시끼가이샤내

(74) 대리인 장수길, 구영창

심사청구 : 없음

(54) **움직임 벡터 검출 방법, 움직임 벡터 산출 프로그램이 기록된 기록 매체, 움직임 검출 장치, 움직임 검출 방법, 화상 엔코딩 장치, 화상 엔코딩 방법, 움직임 벡터 산출 방법, 움직임 벡터 산출 프로그램이 기록된 기록 매체**

요약

(a) 처리될 현재 화상의 블록에 대응하는 참조 화상으로부터 블록을 추출하는 단계 - 참조 화상의 블록의 크기는 현재 화상의 블록의 크기와 동일하고, 참조 화상의 블록의 원점은 현재 화상의 블록의 원점과 일치함-, (b) 소정의 검색 영역에서 참조 화상의 블록을 움직이면서, 현재 화상의 블록과 참조 화상의 블록 간의 잔여분을 얻는 단계, (c) 움직임 벡터를 산출하도록 참조 화상으로부터 최소 잔여분을 갖는 블록을 검출하는 단계, (d) 참조 화상의 블록의 픽셀 데이터 및 현재 화상의 블록의 픽셀 데이터를 직교 변환하는 단계, 및 (e) 참조 화상의 블록의 직교 변환된 데이터와 현재 화상의 각 블록의 직교 변환된 데이터 간의 잔여분을 얻는 단계를 포함하는 움직임 벡터 산출 방법이 개시된다.

대표도

도15

명세서

도면의 간단한 설명

도 1은 종래의 MPEG 엔코더의 구조를 도시하는 블록도.

도 2는 블록 매칭 처리를 설명하기 위한 개략도.

도 3은 종래의 동화상 엔코딩 장치의 구조를 도시하는 개략도.

도 4는 본 발명의 따른 데이터 처리 장치의 구조의 일례를 도시하는 블록도.

도 5는 MPEG2 엔코딩 처리를 설명하기 위한 순서도.

도 6은 본 발명에 따른 움직임 벡터 산출 처리에서의 현 프레임 블록의 처리를 설명하기 위한 개략도.

도 7은 본 발명에 따른 움직임 벡터 산출 처리에서의 현 프레임 블록의 처리를 설명하기 위한 개략도.

도 8은 본 발명에 따른 움직임 벡터 산출 처리에서의 현 프레임 블록의 처리를 설명하기 위한 개략도.

도 9는 지그재그 스캔 처리를 설명하기 위한 개략도.

도 10은 본 발명에 따른 움직임 벡터 산출 처리에서의 참조 프레임의 블록의 처리를 설명하기 위한 개략도.

도 11은 본 발명에 따른 움직임 벡터 산출 처리에서의 참조 프레임의 블록의 처리를 설명하기 위한 개략도.

도 12는 본 발명에 따른 움직임 벡터 산출 처리에서의 참조 프레임의 블록의 처리를 설명하기 위한 개략도.

도 13은 프레임 내 엔코딩 처리 또는 프레임 간 엔코딩 처리가 수행되는지의 여부를 판단하기 위한 함수를 도시한 그래프.

도 14는 프레임 내 엔코딩 처리 또는 프레임 간 엔코딩 처리가 수행되는지의 여부를 판단하기 위한 함수를 도시한 그래프.

도 15는 본 발명에 따른 움직임 벡터 산출 처리를 설명하기 위한 순서도.

도 16은 본 발명에 따른 움직임 벡터 산출 처리를 설명하기 위한 순서도.

도 17은 바둑판 모양으로 솎아내는 처리를 설명하기 위한 개략도.

도 18a 내지 도 18c는 연속적인 데이터와 같은 바둑판 모양의 데이터의 배치를 설명하기 위한 개략도.

도 19a 및 도 19b는 본 발명에 따른 MPEG2 엔코더의 엔코딩 처리를 설명하기 위한 개략도.

도 20은 본 발명에 따른 MPEG2 엔코더의 엔코딩 처리에 사용된 메모리 구조를 설명하기 위한 개략도.

도 21은 본 발명에 따른 MPEG2 엔코더의 엔코딩 처리를 설명하기 위한 타이밍차트.

도 22는 본 발명에 따른 MPEG2 엔코더의 움직임 벡터 산출 처리를 설명하기 위한 순서도.

도 23은 본 발명에 따른 MPEG2 엔코더의 움직임 벡터 산출 처리를 설명하기 위한 순서도.

도 24는 본 발명에 따른 MPEG2 엔코더의 움직임 벡터 산출 처리를 설명하기 위한 순서도.

도 25는 본 발명의 실시예를 설명하기 위한 개략도.

도 26은 본 발명의 실시예를 설명하기 위한 순서도.

도 27은 본 발명에 따른 화상 엔코딩 장치의 구조의 예를 도시한 블록도.

도 28은 화상 엔코딩 장치의 글로벌 벡터 검출부에 사용된 매크로 블록 추출 처리를 설명하기 위한 개략도.

도 29는 하나의 화상을 복수의 영역으로 분할하여 글로벌 벡터를 얻기 위한 처리로서, 화상 엔코딩 장치의 글로벌 벡터 검출부에 의해 수행되는 처리를 설명하기 위한 개략도.

도 30은 화상 엔코딩 장치의 움직임 검출 처리를 설명하기 위한 순서도.

도 31은 본 발명에 따른 화상 엔코딩 장치의 구조를 도시한 블록도.

도 32는 본 발명에 따른 화상 엔코딩 장치의 엔코딩 처리를 설명하기 위한 순서도.

<도면의 주요 부분에 대한 부호의 설명>

101 : 입력 단자

102, 114, 202 : 프레임 메모리

103 : 움직임 벡터 검출 회로

104 : 모드 설정 회로

106, 208 : DCT 처리 회로

107, 209 : 양자화 회로

108 : 가변 길이 엔코딩 회로

109, 211 : 버퍼

110, 212 : 역 양자화 회로

112, 213 : 역 DCT 처리 회로

115, 216 : 움직임 보상 회로

203 : 움직임 검출부

204 : 잔여분 정보 생성 회로

205 : 글로벌 벡터 생성부

206 : 제어부

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 발명은, 예를 들면, MPEG 2 (Moving Picture Experts Group 2) 방법에 따른 엔코딩 처리를 소프트웨어 처리에 의해 수행하는 데 적합한 움직임 벡터 산출 방법에 관한 것이다. 본 발명은 또한 움직임 벡터 산출 프로그램이 기록된 기록 매체에 관한 것이다.

화상을 고효율로 압축하는 방법으로서, MPEG2 방법이 일반적으로 되었다. MPEG2 방법에서, 비디오 신호는 움직임 보상 예측 엔코딩 방법 및 DCT (Discrete Cosine Transform, 이산 코사인 변환) 방법에 의해 압축되고 엔코딩된다.

MPEG2 방법에 있어서, 세가지 타입의 화상들, 즉 I (Intra, 인트라) 화상, P (Predictive, 예측) 화상, 및 B (Bidirectionally Predictive, 양방향 예측) 화상이 전송된다. I 화상에 대해, DCT 엔코딩 처리는 같은 프레임의 픽셀들을 사용하여 수행된다. P 화상에 대해서, DCT 엔코딩 처리는 엔코드된 I 화상 또는 P 화상을 참조하면서 움직임 보상 예측 처리를 사용하여 수행된다. B 화상에 대해서, DCT 엔코딩 처리는 B 화상에 선행하거나 후행하는 I 화상들이나 P 화상들을 참조하면서 움직임 예측 처리를 사용하여 수행된다.

P 화상, 또는 B 화상에 대해서, 인트라-엔코딩 처리는 각각의 매크로 블록에 대해 수행될 수 있다. 즉, 화상이 DC 성분들을 많이 포함하는 경우에, 그 화상은 인트라-엔코딩 처리에 의해 효과적으로 압축될 수 있다. 이에 따라, 이러한 경우에는, 인트라-엔코딩 처리가 수행된다.

도 1은 종래의 MPEG2 엔코더의 구조의 예를 나타내는 블록도이다. 도 1을 참조하면, 디지탈 비디오 신호 성분이 입력 단자(101)에 공급된다. 디지탈 비디오 신호 성분은 휘도 신호 Y와 색차 신호들 Cb 및 Cr을 포함한다. 디지탈 비디오 신호는 입력 단자(101)로부터 움직임 벡터 검출 회로(103)를 통해 프레임 메모리(102)에 공급된다. 프레임 메모리(102)는 디지탈 비디오 신호를 임시로 저장한다. 프레임 메모리(102)는 현재 화상, 과거 참조 화상, 및 미래 참조 화상으로 된 화상들의 적어도 세개의 프레임들에 대한 저장 용량을 갖는다.

움직임 벡터 검출 회로(103)는 프레임 메모리(102)에 저장된 데이터를 사용하여 참조 화상과 현재 화상 사이의 움직임 벡터를 구한다. 움직임 벡터 MV는 각각의 매크로 블록에 대해 구해진다. 각각의 매크로 블록은 예를 들어 (16×16) 픽셀들로 구성된다. 구해진 움직임 벡터 MV는 가변 길이 코드 엔코딩 회로(108)와 움직임 보상 회로(115)에 공급된다. 움직임 벡터 검출 회로(103)는 잔여 정보 e를 모드 설정 회로(104)에 공급한다. 잔여 정보 e는 움직임 벡터 검출 회로에 의해 움직임 벡터 MV에 따라 구해진다.

프레임 메모리(102)의 출력 신호는 움직임 벡터 검출 회로(103)를 통해 스위치 회로(105)의 단자(105A)에 공급된다. 또한, 프레임 메모리(102)의 출력 신호는 감산 회로(106)와 감산 회로(107)에 공급된다. 감산 회로(106)의 출력 신호는 스위치 회로(105)의 단자(105B)에 공급된다. 감산 회로(107)의 출력 신호는 스위치 회로(105)의 단자(105C)에 공급된다.

스위치 회로(105)는 모드 세팅 회로(104)로부터 수신된 모드 설정 신호에 따라 엔코딩 모드를 변화시킨다. 즉, 프레임 내 엔코딩 모드에서는, 스위치 회로(105)가 단자(105A)를 선택한다. 전방 예측 엔코딩 모드에서는, 스위치 회로(105)가 단자(105B)를 선택한다. 양방향 예측 엔코딩 모드에서는, 스위치 회로(105)가 단자(105C)를 선택한다.

I 화상이 전송되면, 프레임 내 예측 엔코딩 처리가 수행된다. 이 시점에서, 스위치 회로(105)는 단자(105A)를 선택한다. 프레임 메모리(102)는 현재 프레임의 화상 데이터를 출력한다. 현재 프레임의 화상 데이터는 스위치 회로(105)를 통해 DCT 회로(106)에 공급된다.

예를 들어, DCT 회로(106)는 시간 영역에서의 (8×8) 픽셀들로 된 각각의 블록의 비디오 신호를 주파수 영역의 스펙트럼 데이터로 변환한다. 스펙트럼 데이터는 지그제그-스케닝되어 낮은 주파수 성분들로부터 (즉, DC 성분들로부터) 높은 주파수 성분들까지 판독된다. DCT 회로(106)의 출력 신호는 양자화 회로(107)에 공급된다.

양자화 스케일은 전송 버퍼(109)로부터 수신된 출력 비트 속도 정보에 따라 양자화 회로(107) 내에 설정된다. 양자화 회로(107)는 양자화 스케일에 따라 DCT 회로(106)로부터 수신된 스펙트럼 데이터를 양자화한다. 이에 따라, 출력 비트 스트림의 비트 속도가 일정하게 유지된다.

양자화 회로(107)의 출력 신호는 가변 길이 코드 엔코딩 회로(108)와 역 양자화 회로(110) 둘다에 공급된다. 가변 길이 코드 엔코딩 회로(108)는 스펙트럼 데이터 (양자화 회로(107)로부터 수신됨), 움직임 벡터, 양자화 스케일, 및 예측 모드를 가변 길이 코드로 엔코딩한다.

가변 길이 코드 엔코딩 회로(108)는 엔코딩된 비트 스트림을 출력한다. 엔코딩된 비트 스트림은 임시로 전송 버퍼(109)에 저장된다. 필요한 비트 속도에 따라 데이터 스트림이 전송 버퍼(109)로부터 판독된다.

상술한 바와 같이, 양자화 회로(107)의 출력 신호는 역 양자화 회로(110)에 공급된다. 역 양자화 회로(110)의 출력 신호는 IDCT 회로(112)에 공급된다. I 화상의 경우, DCT 처리는 같은 프레임의 픽셀들을 사용하여 수행된다. 이에 따라, 역 양자화 회로(110)와 IDCT 회로(112)는 원래의 화상을 형성한다. 하나의 화상에 대한 디지탈 비디오 데이터는 가산 회로(113)를 통해 화상 프레임 메모리(114)에 공급된다. 프레임 메모리(114)에 저장된 데이터는 다음번 P 화상 또는 B 화상의 참조 프레임 데이터로서 사용된다.

P 화상이 전송되는 경우에는, 참조 프레임을 사용하여 순방향 예측 엔코딩 처리가 수행된다. 이 시점에서, 스위치 회로(105)는 단자(105B)를 선택한다. 프레임 메모리(102)는 현재 프레임의 디지탈 비디오 신호를 출력한다. 프레임 메모리(102)의 출력 신호는 감산 회로(106)에 공급된다.

프레임 메모리(114)는 참조 프레임의 데이터를 저장한다. 움직임 보상 회로(115)는 참조 프레임의 데이터의 동작을 보상하고 결과 데이터를 감산 회로(116)에 공급한다. 감산 회로(116)는 현재 프레임의 데이터와 움직임 보상된 참조 프레임의 데이터 간의 차분을 구한다. 차분 데이터는 스위치 회로(105)를 통해 DCT 회로에 공급된다.

DCT 회로(106)는 차분 데이터에 대해 DCT 처리를 수행한다. 즉, DCT 회로(106)는 차분 데이터를 스펙트럼 데이터로 변환한다. DCT 회로(106)의 출력 데이터는 양자화 회로(107)에 공급된다. 양자화 회로(107)는 DCT 회로(106)로부터 수신된 스펙트럼 데이터를 양자화한다.

양자화 회로(107)의 출력 신호는 가변 길이 코드 엔코딩 회로(108)와 역 양자화 회로(110)에 모두 공급된다. 가변 길이 코드 엔코딩 회로(108)는 스펙트럼 데이터 (양자화 회로(107)로부터 수신됨), 움직임 벡터, 양자화 스케일, 및 예측 모드를 가변 길이 코드로써 엔코딩한다.

가변 길이 코드 엔코딩 회로(108)는 엔코딩된 비트 스트림을 출력한다. 엔코딩된 비트 스트림은 전송 버퍼(109)에 임시로 저장된다. 필요한 비트 속도에 따라 데이터 스트림이 전송 버퍼(109)로부터 판독된다. 데이터 스트림이 출력 단자(111)로부터 얻어진다.

양자화 회로(107)의 출력 신호는 역 양자화 회로(110)에 공급된다. 역 양자화 회로(110)의 출력 신호는 IDCT 회로(112)에 공급된다. P 화상의 경우에, 참조 프레임의 데이터와 현재 프레임의 데이터 사이의 차분에 대해 DCT 처리가 수행된다. 이에 따라, 역 양자화 회로(110)와 IDCT 회로(112)는 참조 프레임의 데이터와 현재 화상의 데이터 사이의 차분을 구한다. 차분 데이터는 가산 회로(113)에 공급된다.

프레임 메모리(114)는 참조 화상의 데이터를 움직임 보상 회로(115)를 통해 가산 회로(113)에 공급한다. 가산 회로(113)는 상기 차분 데이터를 참조 화상의 데이터에 가산한다. 가산 회로(113)의 출력 신호는 다음 참조 프레임의 데이터로서 프레임 메모리(114)에 저장된다.

B 화상이 전송되는 경우에는, 과거 참조 프레임과 미래 참조 프레임을 참조하여, 양방향 예측 엔코딩 처리가 수행된다. 이러한 경우에, 스위치 회로(105)는 단자(105C)를 선택한다. 프레임 메모리(102)는 현재 프레임의 디지탈 비디오 신호를 출력한다. 프레임 메모리(102)의 출력 신호는 감산 회로(117)에 공급된다.

프레임 메모리(114)는 과거 참조 프레임의 데이터와 미래 참조 프레임의 데이터를 저장한다. 움직임 보상 회로(115)는 상기 두 참조 프레임들의 데이터를 움직임 보상한다. 움직임 보상 회로(115)의 출력 신호는 감산 회로(117)에 공급된다. 감산 회로(117)는 현재 프레임의 데이터와 움직임 보상된 과거 참조 프레임의 데이터 및 미래 참조 프레임 데이터 사이의 차분을 구한다. 차분 데이터는 스위치 회로(105)를 통해 DCT 회로(106)에 공급된다.

DCT 회로(106)는 차분 데이터에 대해 DCT 처리를 수행한다. 즉, DCT 회로(106)는 차분 데이터를 스펙트럼 데이터로 변환한다. DCT 회로(106)의 출력 신호는 양자화 회로(107)에 공급된다. 양자화 회로(107)는 DCT 회로(106)로부터 수신된 스펙트럼 데이터를 양자화한다.

양자화 회로(107)의 출력 신호는 가변 길이 코드 엔코딩 회로(108)와 역 양자화 회로(110)에 모두 공급된다. 가변 길이 코드 엔코딩 회로(108)는 스펙트럼 데이터 (양자화 회로(107)로부터 수신됨), 움직임 벡터, 양자화 스케일, 및 예측 모드를 가변 길이 코드를 사용하여 엔코드한다.

가변 길이 코드 엔코딩 회로(108)는 엔코딩된 비트 스트림을 출력한다. 비트 스트림은 전송 버퍼(109)에 임시로 저장된다. 전송 버퍼(109)는 필요한 비트 속도에 따라 데이터 스트림을 판독한다. 데이터 스트림은 출력 단자(111)로부터 얻어진다.

최근에, CPU(Central Processing Units)들의 처리 속도가 매우 빨라졌고 큰 저장 용량의 메모리들이 저렴해졌기 때문에, 상술한 MPEG2 엔코더가 소프트웨어에 의해 수행될 수 있다.

그러나, MPEG2 엔코딩 처리에서, 움직임 벡터를 산출하는 처리가 요구된다. 움직임 벡터는 블록 매칭 처리에 의해 얻어진다. 즉, 처리될 현재 프레임으로부터 분할된 블록과 같은 크기 및 같은 기점(origin)을 갖는 블록이 참조 프레임으로부터 추출된다. 참조 프레임의 블록이 선정된 검색 영역에서 이동되면서, 참조 프레임의 블록의 픽셀들과 현재 프레임의 관련 블록의 픽셀들 사이의 차분의 절대값들의 합이 잔여분(residual)으로서 구해진다. 최소의 잔여분을 갖는 참조 프레임의 블록이 구해진다. 블록 매칭 처리는 많은 산출 단계들을 필요로 하기 때문에, MPEG2 엔코딩 처리를 소프트웨어로 수행하기가 어렵다.

즉, 도 2에 도시된 현재 프레임(401)의 블록 CBLK의 움직임 벡터가 구해지면, 블록 CBLK를 기정으로 이에 대응하는 참조 프레임(402)의 블록 RBLK의 주변에서 검색 영역 SA가 한정된다. 참조 프레임의 상기 블록 RBLK는 검색 영역 SA으로부터 추출된다. 기준 블록 RBLK의 픽셀들 (16×16)과 현재 블록 CBLK의 픽셀들 (16×16) 사이의 차분이 구해진다. 이 차분들의 절대값들의 합이 잔여분으로서 구해진다. 참조 프레임(402)의 블록 RBLK가 선정된 검색 영역 SA 내에서 이동된다. 검색 영역 SA 내의 블록 RBLK의 각각의 위치에서, 블록 RBLK의 픽셀들과 블록 CBLK의 픽셀들 사이의 차분들이 구해진다. 이 차분들의 절대값들의 합이 잔여분으로서 구해진다. 구해진 합들이 비교된다. 최소의 잔여분을 갖는 블록이 매칭된 블록으로서 처리된다. 매칭된 블록을 사용하여, 움직임 벡터가 구해진다.

블록 매칭 처리에 의해 움직임 벡터를 검출하기 위해, 각각의 블록이 (16×16) 픽셀들로 구성되는 경우, 픽셀들의 차분들을 구하는데 16×16 =256회의 감산 연산들이 필요하다. 차분들의 절대값들의 합을 구하기 위해서는, 256회의 가산 연산들이 필요하다.

선정된 검색 영역에서 기준 블록을 한 픽셀 씩 움직임에 의해 움직임 벡터가 검출되는 경우, 잔여분들은 검색 영역 내의 픽셀들의 개수에 대응하는 횟수만큼 구해져야 한다. 이에 따라, 블록을 선정된 검색 영역 내에서 1 픽셀 스텝으로 움직

이고, 움직임 벡터가 최소의 잔여분을 갖는 블록의 위치로써 검출됨으로써 잔여분들이 구해지는 경우, 산출 단계들의 횟수는 엄청나게 커진다. 이에 따라, 소프트웨어에 의해 MPEG2 엔코딩 처리를 수행하기란 어렵다.

고속으로 움직임 벡터를 검색하기 위해, 두 개의 방법들이 고려될 수 있다. 첫번째 방법으로는, 블록 매칭 처리가 수행될 때마다, 산출 단계들의 횟수가 감소된다. 두번째 방법으로는, 검색 영역에서의 블록 매칭 처리의 횟수가 감소된다.

첫번째 방법의 예에 따르면, 참조 프레임의 블록의 픽셀들과 현재 프레임의 관련된 블록의 픽셀들 사이의 차분들의 절대값들의 합이 산출되면서, 이 합이 선정된 임계값과 비교된다. 상기 합이 상기 선정된 임계값보다 더 크면, 처리는 중단된다.

움직임 벡터가 참조 프레임의 블록의 픽셀들과 현재 프레임의 관련 블록의 픽셀들 사이의 차분들의 절대값들의 합의 최소값을 구함으로써 구해지기 때문이다. 이에 따라, 상기 합이 선정된 임계값을 초과하면, 최소값이 안된다. 이에 따라, 상기 처리를 계속할 의미가 없다. 결과적으로, 합이 상기 선정된 임계값보다 큰 경우, 처리는 중단된다. 그 결과, 산출 단계들의 횟수가 감소될 수 있고, 움직임 벡터가 고속으로 검출될 수 있다.

그러나, 이 경우에, 이러한 임계값이 할당되는 것이 어렵다. 임계값이 너무 작으면, 모든 지점들에서 중단된다. 이에 따라 움직임 벡터가 정확하게 검출될 수 없다. 반면에, 임계값이 너무 크면, 처리가 중단되지 않기 때문에, 처리의 효율이 향상될 수 없다.

블록 매칭 처리에 대한 산출 단계들의 개수를 감소시키기 위해, 참조 프레임의 픽셀들과 현재 프레임의 픽셀들을 바둑판 모양으로 솎아내는 방법이 있다. 이러한 경우에, 차분들의 절대값들의 합을 산출하는 연산 횟수가 반으로 줄어들 수 있다.

그러나, 블록들이 바둑판 모양으로 솎아지면, 픽셀들의 데이터의 연속성이 상실되고, MMX 명령들이 사용될 수 없다. MMX 명령은 복수의 연속 데이터 부분들이 한 번에 처리되도록 해주기 때문에, 최근의 개인용 컴퓨터들에는 MMX 기능을 처리하는 CPU들이 장착된다. 블록 매칭 처리가 MMX 명령을 사용하여 픽셀들 사이의 차분들의 절대값들의 합을 구하기 때문에, 블록 매칭 처리는 고속으로 수행될 수 있다. 그러나, 블록들의 픽셀들이 바둑판 모양으로 솎아지면, 픽셀들의 데이터의 연속성이 상실되기 때문에, MMX 명령이 사용될 수 없다. 이에 따라, 블록들의 픽셀들이 바둑판 모양으로 솎아지고 이에 따라 블록 매칭 처리의 횟수가 감소되더라도, 처리 시간이 현저하게 줄어들 수 없다. 움직이는 화상의 화상 데이터를 엔코딩하는 종래의 동화상 엔코딩 장치에서는, 움직임 보상된 프레임 간 예측 엔코딩 처리와 DCT (이산 코사인 변환) 처리의 조합인 하이브리드 엔코딩 처리가 (8×8) 픽셀들로 된 각각의 블록에 대해 수행된다.

동화상 엔코딩 장치는 인접한 프레임들 사이의 움직임 벡터를 검출하고 검출된 움직임 벡터를 사용하여 엔코딩된 데이터의 양을 감소시키도록 움직임 보상한다.

종래의 동화상 엔코딩 장치는 다양한 방법들로 화상의 동작을 검출한다. 화상의 동작이 검출되면, 가끔 인접한 프레임들의 연관된 위치의 매크로 블록들이 비교된다. 이에 따라, 인접한 프레임들의 연관된 위치 주위의 선정된 영역이 휘도값들의 차분이 작은 매크로 블록들을 찾기 위해 검색된다.

동화상을 찍는 카메라가 선정된 위치에 고정되고, 대상물이 움직이는 경우, 대상물이 움직이는 방향은 전체 화상의 각 위치에서 변화한다. 이에 따라, 시작점 주위의 매크로 블록들이 검색된다. 대상물의 동작이 큰 경우, 상기 방향은 검색 영역을 벗어날 수 있다. 이러한 경우에, 프레임 내 엔코딩 처리가 프레임 간 엔코딩 처리를 대신하여 수행된다.

종래의 동화상 장치에서는, 화상의 동작을 검출하기 위한 산출 단계의 개수를 줄이기 위해 계층적 검색 방법이 제안되었다. 이 방법에서는, 화상의 동작이 검출되어, 하나의 움직임 벡터가 복수개의 매크로 블록들을 사용하여 검출된다. 각각의 매크로 블록의 움직임 벡터를 구하기 위해, 복수의 매크로 블록들에 대한 움직임 벡터가 사용된다.

그러나, 이러한 동화상 엔코딩 장치에 있어서, 차분이 작은 매크로 블록들을 검출하기 위해 매크로 블록들이 선정된 영역 내에서 검색된다. 이에 따라, 크게 움직이는 대상물이 처리되는 경우, 검색 영역을 넓히는 것이 필요하다. 그러므로,

화상의 동작을 검출하는데 필요한 처리 시간이 지수 함수적으로 증가한다.

동화상 엔코딩 장치에 있어서, 대상물의 동작이 큰 경우, 프레임 내 엔코딩 처리 (같은 프레임의 매크로 블록들에 대해)가 프레임 간 엔코딩 처리 (복수개의 프레임들의 매크로 블록들)를 대신하여 수행된다. 이 경우에, 촬영된 화상의 동작이 선정된 검색 영역을 초과하는 경우, 프레임 내 엔코딩 처리가 모든 매크로 블록들에 대해 수행된다면, 산출 단계들의 개수가 증가한다. 이에 따라, 장치의 부하가 크게 된다. 이러한 상황은 전체 화상이 검색 영역을 초과하여 움직이면 대상물이 패닝되는 경우에 발생한다.

복수의 매크로 블록들의 움직임 벡터들을 구하는 동화상 엔코딩 장치에 있어서, 화상이 스크린을 초과하여 크게 움직이는 경우, 움직임 벡터가 검출될 수 없다. 이에 따라, 복수의 매크로 블록들의 움직임 벡터들이 검출될 수 없다.

도 3은, 움직임 보상 프레임 간 예측 처리와 DCT(Discrete Cosine Transform) 처리를 조합한 하이브리드 엔코딩 처리에 의해 동화상의 화상 데이터를 엔코딩하는 종래의 동화상 엔코딩 장치의 구조의 일례를 도시하고 있다.

도 3에서, 입력 MB(Macro Block) 신호 S11가 단자(501)에 공급된다. 움직임 벡터 신호 MV가 매크로 블록으로서 하나씩 단자(502)에 공급된다. 입력 MB 신호 S511과 움직임 벡터 신호 MV는 움직임 보상 회로(503)에 공급된다.

움직임 보상 회로(503)는 내부 화상 메모리를 구비한다. 예측 화상 신호(이하 예측 MB 신호로 칭함)이 매크로 블록 MB로서 움직임 벡터 신호 MV에 대응하는 화상 메모리로부터 하나씩 판독된다. 움직임 보상 회로(503)는 움직임 벡터 신호 MV로부터 얻어진 예측 MB 신호인 신호 S512를 출력한다.

산출 디바이스(504)는 가산 신호인 입력 MB 신호 S511와 감산 신호인 신호 S512를 매크로 블록으로서 하나씩 가산한다. 따라서, 산출 디바이스(504)는 입력 MB 신호와 신호 S512 간의 차분을 산출하여, 그 차분을 예측 잔여 MB 신호 S513로서 출력한다.

예측 잔여 MB 신호 S513이 DCT 회로에 공급된다. DCT 회로(505)는, 예측 잔여 MB 신호 S513의 8×8 픽셀 블록 각각에 대하여 2차원 DCT 처리를 수행하여, DCT 계수를 출력한다. DCT 계수는 양자화 회로(506)에 공급된다.

양자화 회로(506)는 단자(507)로부터 수신된 양자화 스케일 mQ, 움직임 보상 회로(503)로부터 수신된 예측 MB 신호의 절대치의 차분, 및 절대치와 평균치 간의 차분에 대응하여 DCT 계수를 양자화하여, 그 결과를 양자화 신호로서 출력한다.

양자화 회로(506)로부터 수신된 양자화 신호와 그에 대응하는 움직임 벡터 MV가 가변 길이 코드 엔코딩(VLC) 회로(508)에 공급된다. 가변 길이 엔코딩 회로(508)는 양자화 신호와 움직임 벡터 MV를 MPEG 신택스에 대응하는 가변 길이 코드로 엔코딩한다.

가변 길이 엔코딩 회로(508)의 출력 신호는 버퍼 메모리(508)에 공급된다. 버퍼 메모리(509)는, 단기간에 생성되고 가변 길이 코드 엔코딩 회로(508)로부터 수신된 데이터의 비트 수 변동을 완화시키고, 엔코딩된 비트 스트림을 원하는 비트 속도로 출력한다. 버퍼 메모리(509)로부터 수신되어 엔코딩된 비트 스트림은, 단자(510)로부터 출력된다.

양자화 회로(511)로부터 수신된 양자화 신호와 양자화 스케일이 역 양자화 회로(511)에 공급된다. 역 양자화 회로(511)는 양자화 스케일에 따라 양자화 신호를 역으로 양자화한다. 역 양자화 회로(511)의 출력 신호가 역 DCT 회로(512)에 공급된다. 역 DCT 회로(512)는 역 양자화 회로(511)로부터 수신된 신호에 대해 역 DCT 처리를 수행하여, 그 결과 신호를 예측 잔여 MB 신호 S515로서 산출 디바이스(513)에 출력한다.

산출 디바이스(513)는 산출 디바이스(504)에 공급된 예측 MB 신호(504)도 수신한다. 이러한 화상 신호는 수신기 측(디코더 측)의 출력 신호와 동일하다.

종래의 동화상 엔코딩 장치는 단자(501)로부터 수신된 모든 화상에 대해 DCT 처리 및 양자화 처리를 수행한다. 동작 화상 엔코딩 장치는, 화상 데이터에 대해 DCT 처리 및 양자화 처리를 수행하고 DCT 계수에 대한 모든 산출 단계를 종료한 후, 엔코딩될 화상의 매크로 블록 각각의 DCT 계수가 존재하는지의 여부를 판정한다.

그러나, 동화상 엔코딩 장치는, DCT 계수가 최종적으로 '0'이 된 경우에도 모든 매크로 블록에 대해 산출 단계를 수행하기 때문에, 불필요한 단계가 수행된다.

게다가, 종래의 동화상 엔코딩 장치는, 모든 산출 단계를 수행한 후에만 매크로 블록의 모든 DCT 계수가 '0'인지를 판정하기 때문에, 모든 산출 단계가 수행되어야만 한다.

발명이 이루고자하는 기술적 과제

본 발명은 상기와 같은 관점에서 이루어졌다.

본 발명의 제1 목적은 움직임 벡터를 검출하기 위한 블록 매칭 처리의 산출 단계 수를 감소시켜서 고속으로 검출하는 것이다.

본 발명의 제2 목적은, 선정된 검색 영역 내의 블록 매칭 처리의 회수가 감소되어, 처리 속도를 증가시키고 MMX 명령이 효율적으로 사용될 수 있는 움직임 벡터 산출 방법 및 그 프로그램에 기록되어 있는 기록 매체를 제공하는 것이다.

본 발명의 제3 목적은 스크린 전체 상에서 크게 동작하는 화상의 움직임 벡터가 검출될 수 있게 하는 움직임 검출 장치 및 움직임 검출 방법을 제공하는 것이다.

본 발명의 제4 목적은 DCT 계수가 최종적으로 '0'이 되는 화상에 대한 엔코딩 처리 기간이 단축될 수 있게 하는 화상 엔코딩 장치 및 화상 엔코딩 방법을 제공하는 것이다.

본 발명의 제1 양태는, (a) 처리될 현재 화상의 블록에 대응하는 참조 화상으로부터 블록을 추출하는 단계 - 상기 참조 화상의 블록의 크기는 상기 현재 화상의 블록의 크기와 동일하고, 상기 참조 화상의 블록의 원점은 상기 현재 화상의 블록의 원점과 일치함-, (b) 소정의 검색 영역에서 상기 참조 화상의 블록을 움직이면서, 상기 현재 화상의 블록과 상기 참조 화상의 블록 간의 잔여분을 얻는 단계, (c) 움직임 벡터를 산출하도록 상기 참조 화상으로부터 최소 잔여분을 갖는 블록을 검출하는 단계, (d) 상기 참조 화상의 블록의 픽셀 데이터 및 상기 현재 화상의 블록의 픽셀 데이터를 직교 변환하는 단계, 및 (e) 상기 참조 화상의 블록의 직교 변환된 데이터와 상기 현재 화상의 각 블록의 직교 변환된 데이터 간의 잔여분을 얻는 단계를 포함하는 움직임 벡터 산출 방법이다.

본 발명의 제2 양태는, 움직임 벡터 산출 프로그램이 기록된 기록 매체로서, 상기 움직임 벡터 산출 프로그램은 상기 기록 매체를 갖는 시스템으로 하여금 다음의 단계를 수행하게 하는 기록 매체에 있어서, 다음의 단계는 (a) 처리될 현재 화상의 블록에 대응하는 참조 화상으로부터 블록을 추출하는 단계 -상기 참조 화상의 블록의 크기는 상기 현재 화상의 블록의 크기와 동일하고, 상기 참조 화상의 블록의 원점은 상기 현재 화상의 블록의 원점과 일치함-, (b) 소정의 검색 영역에서 상기 참조 화상의 블록을 움직이면서, 상기 현재 화상의 블록과 상기 참조 화상의 블록 간의 잔여분을 얻는 단계, (c) 움직임 벡터를 산출하도록 상기 참조 화상으로부터 최소 잔여분을 갖는 블록을 검출하는 단계, (d) 상기 참조 화상의 블록의 픽셀 데이터 및 상기 현재 화상의 블록의 픽셀 데이터를 직교 변환하는 단계, 및 (e) 상기 참조 화상의 블록의 직교 변환된 데이터와 상기 현재 화상의 각 블럭의 직교 변환된 데이터 간의 잔여분을 얻는 단계를 포함하는 기록 매체이다.

본 발명의 제3 양태는, (a) 처리될 현재 화상의 블록에 대응하는 참조 화상으로부터 블록을 추출하는 단계 - 상기 참조 화상의 블록의 크기는 상기 현재 화상의 블록의 크기와 동일하고, 상기 참조 화상의 블록의 원점은 상기 현재 화상의 블록의 원점과 일치함-, (b) 소정의 검색 영역에서 상기 참조 화상의 블록을 움직이면서, 상기 현재 화상의 블록과 상기 참조 화상의 블록 간의 잔여분을 얻는 단계, (c) 움직임 벡터를 산출하도록 상기 참조 화상으로부터 최소 잔여분을 갖는 블록을 검출하는 단계, (d) 상기 참조 화상의 블록의 픽셀과 상기 현재 화상의 블록의 픽셀 간의 잔여분을 산출하면서, 상기 얻어진 잔여분과 소정의 임계값을 비교하는 단계, (e) 상기 잔여분이 상기 소정의 임계값보다 큰 경우, 상기 움직임 벡터의 산출을 중지하는 단계, 및 (f) 화상의 특성에 대응하는 상기 소정의 임계값의 초기값을 설정하는 단계를 포함하는 움직임 벡터 산출 방법이다.

본 발명의 제4 양태는, 움직임 벡터 산출 프로그램이 기록된 기록 매체로서, 상기 움직임 벡터 산출 프로그램은 상기 기록 매체를 갖는 시스템으로 하여금 다음의 단계를 수행하게 하는 기록 매체에 있어서, 다음의 단계는 (a) 처리될 현재 화상의 블록에 대응하는 참조 화상으로부터 블록을 추출하는 단계 - 상기 참조 화상의 블록의 크기는 상기 현재 화상의 블록의 크기와 동일하고, 상기 참조 화상의 블록의 원점은 상기 현재 화상의 블록의 원점과 일치함-, (b) 소정의 검색 영역에서 상기 참조 화상의 블록을 움직이면서, 상기 현재 화상의 블록과 상기 참조 화상의 블록 간의 잔여분을 얻는 단계, (c) 움직임 벡터를 산출하도록 상기 참조 화상으로부터 최소 잔여분을 갖는 블록을 검출하는 단계, (d) 상기 참조 화상의 블록의 픽셀과 상기 현재 화상의 블록의 픽셀 간의 잔여분을 산출하면서, 상기 얻어진 잔여분과 소정의 임계값을 비교하는 단계, (e) 상기 잔여분이 상기 소정의 임계값보다 큰 경우, 상기 움직임 벡터의 산출을 중지하는 단계, 및 (f) 화상의 특성에 대응하는 상기 소정의 임계값의 초기값을 설정하는 단계를 포함하는 기록 매체이다.

본 발명의 제5 양태는, 화상으로부터 복수의 매크로 블록을 추출하기 위한 추출 수단, 상기 추출 수단에 의해 추출된 상기 복수의 매크로 블록 각각의 움직임 벡터를 검출하기 위한 제1 움직임 검출 수단, 상기 제1 움직임 검출 수단에 의해 검출된 개개의 매크로 블록의 움직임 벡터를 갖는 전체 화상의 움직임 벡터를 산출하기 위한 움직임 산출 수단, 및 상기 움직임 산출 수단에 의해 산출된 움직임 벡터를 갖는 각 매크로 블록의 움직임 벡터를 산출하기 위한 제2 움직임 검출 수단을 포함하는 움직임 검출 장치이다.

본 발명의 제6 양태는, (a) 화상으로부터 복수의 매크로 블록을 추출하는 단계, (b) 추출된 상기 복수의 매크로 블록 각각의 움직임 벡터를 검출하는 단계, (c) 검출된 개개의 매크로 블록의 움직임 벡터를 갖는 전체 화상의 움직임 벡터를 산출하는 단계, 및 (d) 산출된 움직임 벡터를 갖는 각 매크로 블록의 움직임 벡터를 산출하는 단계를 포함하는 움직임 검출 방법이다.

본 발명의 제7 양태는, 입력 화상 데이터의 소정의 픽셀 블록의 움직임 벡터를 검출하고 움직임 잔여분 정보를 발생시키기 위한 움직임 검출 수단, 상기 움직임 검출 수단으로부터 수신된 움직임 잔여분 정보를 소정값과 비교하고 판정된 결과를 발생시키기 위한 판정 수단, 화상 데이터에 대해 소정의 처리를 수행하기 위한 화상 데이터 처리 수단 - 상기 소정의 처리는 엔코딩 처리를 위해 요구됨-, 화상 데이터에 대해 상기 엔코딩 처리를 수행하기 위한 엔코딩 수단, 및 상기 판정 수단의 판정 결과에 대응하는 상기 화상 데이터 처리 수단에 의해 수행된 소정의 처리를 건너띄고 상기 엔코딩 수단으로 하여금 상기 엔코딩 처리를 수행하게 하는 제어 수단을 포함하는 화상 엔코딩 장치이다.

본 발명의 제8 양태는, (a) 입력 화상 데이터의 소정의 픽셀 블록의 움직임 벡터를 검출하고 움직임 잔여분 정보를 발생시키는 단계, (b) 상기 움직임 잔여분 정보를 소정값과 비교하고 판정된 결과를 발생시키는 단계, (c) 화상 데이터에 대해 소정의 처리를 수행하는 단계 - 상기 소정의 처리는 엔코딩 처리를 위해 요구됨-, (d) 상기 판정 결과에 대응하는 상기 소정의 처리를 건너띄고 상기 화상 데이터에 대해 상기 엔코딩 처리를 수행하는 단계를 포함하는 화상 엔코딩 방법이다.

본 발명의 제9 양태는, (a) 처리될 현재 화상의 블록에 대응하는 참조 화상으로부터 블록을 추출하는 단계 - 상기 참조 화상의 블록의 크기는 상기 현재 화상의 블록의 크기와 동일하고, 상기 참조 화상의 블록의 원점은 상기 현재 화상의 블록의 원점과 일치함-, (b) 소정의 검색 영역에서 상기 참조 화상의 블록을 움직이면서, 상기 현재 화상의 블록과 상기 참조 화상의 블록 간의 잔여분을 얻는 단계, (c) 움직임 벡터를 산출하도록 상기 참조 화상으로부터 최소 잔여분을 갖는 블록을 검출하는 단계, (d) 상기 현재 화상의 N개의 픽셀과 상기 참조 화상의 N개의 픽셀을 한 번에 추출하는 단계(여기서 N은 정수), (e) 상기 현재 화상의 N개의 픽셀과 상기 참조 화상의 N개의 픽셀을 연속하는 데이터로서 메모리에 저장하는 단계, 및 (f) 잔여분을 얻도록 상기 메모리로부터 상기 현재 화상의 블록의 픽셀과 상기 참조 화상의 블록의 픽셀을 연속하는 데이터로서 판독하는 단계를 포함하는 움직임 벡터 산출 방법이다.

본 발명의 제10 양태는, 움직임 벡터 산출 프로그램이 기록된 기록 매체로서, 상기 움직임 벡터 산출 프로그램은 상기 기록 매체를 갖는 시스템으로 하여금 다음의 단계를 수행하게 하는 기록 매체에 있어서, 다음의 단계는 (a) 처리될 현재 화상의 블록에 대응하는 참조 화상으로부터 블록을 추출하는 단계 - 상기 참조 화상의 블록의 크기는 상기 현재 화상의 블록의 크기와 동일하고, 상기 참조 화상의 블록의 원점은 상기 현재 화상의 블록의 원점과 일치함-, (b) 소정의 검색 영역에서 상기 참조 화상의 블록을 움직이면서, 상기 현재화상의 블록과 상기 참조 화상의 블록 간의 잔여분을 얻는 단계,

(c) 움직임 벡터를 산출하도록 상기 참조 화상으로부터 최소 잔여분을 갖는 블록을 검출하는 단계, (d) 상기 현재 화상의 N개의 픽셀과 상기 참조 화상의 N개의 픽셀을 한 번에 추출하는 단계(여기서 N은 정수), (e) 상기 현재 화상의 N개의 픽셀과 상기 참조 화상의 N개의 픽셀을 연속하는 데이터로서 메모리에 저장하는 단계, 및 (f) 잔여분을 얻도록 상기 메모리로부터 상기 현재 화상의 블록의 픽셀과 상기 참조 화상의 블록의 픽셀을 연속하는 데이터로서 판독하는 단계를 포함하는 기록 매체이다.

본 발명의 제11 양태는, (a) 처리될 현재 화상의 블록에 대응하는 참조 화상으로부터 블록을 추출하는 단계 - 상기 참조 화상의 블록의 크기는 상기 현재 화상의 블록의 크기와 동일하고, 상기 참조 화상의 블록의 원점은 상기 현재 화상의 블록의 원점과 일치함-, (b) 소정의 검색 영역에서 상기 참조 화상의 블록을 움직이면서, 상기 현재 화상의 블록과 상기 참조 화상의 블록 간의 잔여분을 얻는 단계, (c) 대강의 움직임 벡터를 산출하도록 상기 참조 화상으로부터 최소 잔여분을 갖는 블록을 검출하는 단계, (d) 단계(c)에서 얻어진 상기 대강의 움직임 벡터 근처에서 상기 참조 화상의 블록을 움직이면서, 상기 현재 화상의 블록과 상기 참조 화상의 블록 간의 잔여분을 얻는 단계, (e) 미세한 움직임 벡터를 검출하도록 상기 참조 화상으로부터 최소 잔여분을 갖는 블록을 검출하는 단계, (f) 상기 현재 화상의 픽셀과 상기 참조 화상의 픽셀을 제1 메모리에 저장하는 단계, (g) 상기 현재 화상의 N개의 픽셀과 상기 참조 화상의 N개의 픽셀을 한 번에 추출하는 단계(여기서 N은 정수), 및 (h) 상기 현재 화상의 N개의 픽셀과 상기 참조 화상의 N개의 픽셀을 연속하는 데이터로서 제2 메모리에 저장하는 단계를 포함하고, 단계(c)는 상기 제2 메모리에 연속하는 데이터로서 저장된 상기 현재 화상의 N개의 픽셀과 상기 참조 화상의 N개의 픽셀로 수행되고, 단계(e)는 상기 제1 메모리에 저장된 상기 현재 화상의 픽셀과 상기 참조 화상의 픽셀로 수행되는 움직임 벡터 산출 방법이다.

본 발명의 제12 양태는, 움직임 벡터 산출 프로그램이 기록된 기록 매체로서, 상기 움직임 벡터 산출 프로그램은 상기 기록 매체를 갖는 시스템으로 하여금 다음의 단계를 수행하게 하는 기록 매체에 있어서, 다음의 단계는 (a) 처리될 현재 화상의 블록에 대응하는 참조 화상으로부터 블록을 추출하는 단계 - 상기 참조 화상의 블록의 크기는 상기 현재 화상의 블록의 크기와 동일하고, 상기 참조 화상의 블록의 원점은 상기 현재 화상의 블록의 원점과 일치함-, (b) 소정의 검색 영역에서 상기 참조 화상의 블록을 움직이면서, 상기 현재 화상의 블록과 상기 참조 화상의 블록 간의 잔여분을 얻는 단계, (c) 대강의 움직임 벡터를 산출하도록 상기 참조 화상으로부터 최소 잔여분을 갖는 블록을 검출하는 단계, (d) 단계(c)에서 얻어진 상기 대강의 움직임 벡터 근처에서 상기 참조 화상의 블록을 움직이면서, 상기 현재 화상의 블록과 상기 참조 화상의 블록 간의 잔여분을 얻는 단계, (e) 미세한 움직임 벡터를 검출하도록 상기 참조 화상으로부터 최소 잔여분을 갖는 블록을 검출하는 단계, (f) 상기 현재 화상의 픽셀과 상기 참조 화상의 픽셀을 제1 메모리에 저장하는 단계, (g) 상기 현재 화상의 N개의 픽셀과 상기 참조 화상의 N개의 픽셀을 한 번에 추출하는 단계(여기서 N은 정수), 및 (h) 상기 현재 화상의 N개의 픽셀과 상기 참조 화상의 N개의 픽셀을 연속하는 데이터로서 제2 메모리에 저장하는 단계를 포함하고, 단계(c)는 상기 제2 메모리에 연속하는 데이터로서 저장된 현재 화상의 N개의 픽셀과 상기 참조 화상의 N개의 픽셀로 수행되고, 단계(e)는 상기 제1 메모리에 저장된 현재 화상의 픽셀과 상기 참조 화상의 픽셀로 수행되는 기록 매체이다.

본 발명의 제13 양태는, (a) 처리될 현재 화상의 블록에 대응하는 참조 화상으로부터 블록을 추출하는 단계 - 상기 참조 화상의 블록의 크기는 상기 현재 화상의 블록의 크기와 동일하고, 상기 참조 화상의 블록의 원점은 상기 현재 화상의 블록의 원점과 일치함-, (b) 소정의 검색 영역에서 상기 참조 화상의 블록을 움직이면서, 상기 현재 화상의 블록과 상기 참조 화상의 블록 간의 잔여분을 얻는 단계, (c) 움직임 벡터를 산출하도록 상기 참조 화상으로부터 최소 잔여분을 갖는 블록을 검출하는 단계, 및 (d) 상기 참조 화상의 블록의 윤곽 픽셀과 상기 현재 화상의 블록의 윤곽 픽셀을 비교하여 이들 사이의 잔여분을 얻는 단계를 포함하는 움직임 벡터 산출 방법이다.

본 발명의 제14 양태는, 움직임 벡터 산출 프로그램이 기록된 기록 매체로서, 상기 움직임 벡터 산출 프로그램은 상기 기록 매체를 갖는 시스템으로 하여금 다음의 단계를 수행하게 하는 기록 매체에 있어서, 다음의 단계는 (a) 처리될 현재 화상의 블록에 대응하는 참조 화상으로부터 블록을 추출하는 단계 - 상기 참조 화상의 블록의 크기는 상기 현재 화상의 블록의 크기와 동일하고, 상기 참조 화상의 블록의 원점은 상기 현재 화상의 블록의 원점과 일치함-, (b) 소정의 검색 영역에서 상기 참조 화상의 블록을 움직이면서, 상기 현재 화상의 블록과 상기 참조 화상의 블록 간의 잔여분을 얻는 단계, (c) 움직임 벡터를 산출하도록 상기 참조 화상으로부터 최소 잔여분을 갖는 블록을 검출하는 단계, 및 (d) 상기 참조 화상의 블록의 윤곽 픽셀과 상기 현재 화상의 블록의 윤곽 픽셀을 비교하여 이들 사이의 잔여분을 얻는 단계를 포함하는 기록 매체이다.

본 발명의 여타의 목적, 특성과 이점은, 그들의 최상의 실시예에 관한 하기의 상세한 설명을 첨부된 도면을 참조하여 숙지함으로써 더욱 명백해질 것이다.

발명의 구성 및 작용

다음으로, 첨부 도면을 참조하여, 본 발명의 실시예들을 설명한다. 도 4는 본 발명의 제1 실시예에 따른 데이터 처리 장치의 구조의 예를 도시한 블록도이다.

도 4를 참조하면, 참조 부호 1은 CPU(Central Processing Unit)이다. 참조 부호 2는 ROM(Read Only Memory)이다. 참조 부호 3은 RAM(Random Access Memory)이다. CPU(1), ROM(2) 및 RAM(3)은 프로세서 버스(4)에 접속된다.

CPU(1)는 예를 들어 MMX 기능을 갖는 프로세서이다. MMX 기능은 동화상 재생 처리, 화상 편집 처리, 및 사운드 신시사이징 처리 등이 고속으로 수행되게 한다. SIMD(Single Instruction Multiple Data) 기술을 사용하는 MMX 명령에 의해, 연속적인 데이터에 대해 동일한 처리가 한번에 수행될 수 있다.

ROM(2)은 부트 스트랩(boot strap) 프로그램을 저장한다. RAM(3)은 작업 영역으로서 주 메모리이다. RAM(3)의 저장 용량은 예를 들어 64 MB 이상인 것이 좋다.

CPU(1)는 브리지 회로(5)에 접속된다. 브리지 회로(5)는 프로세서 버스(4)에 접속된다. 브리지 회로(5)는 PCI(Peripheral Component Interconnect) 버스(6)에 접속된다. 브리지 회로(5)는 CPU(1), 프로세서 버스(4) 및 PCI 버스(6)를 접속한다.

PCI 버스(6)는 IDE(Integrated Device Electronics) 제어기(7), SCSI(Small Computer System Interface) 제어기(8), 그래픽 가속기(9) 및 IEEE(Institute of Electrical and Electronics Engineers) 1394 제어기(10)에 접속된다.

IDE 제어기(7)는 하드 디스크 드라이브 또는 CD 드라이브와 같은 저장 장치(11)에 접속된다. SCSI 제어기(8)는 하드 디스크 드라이브 또는 CD 드라이브와 같은 저장 장치(12)에 접속된다. SCSI 제어기(8)는 저장 장치뿐만 아니라 화상 스캐너와 같은 주변 장치에도 접속된다. 그래픽 가속기(9)는 디스플레이(13)에 접속된다. IEEE 1394 가속기(9)는 디스플레이(13)에 접속된다. IEEE 1394 제어기(10)는 디지털 VCR(Video Cassette Recorder)과 같은 디지털 오디오 비디오 장치에 접속된다.

PCI 버스(6)는 브리지 회로(14)를 통해 ISA(Industrial Standard Architecture) 버스(15)에 접속된다. 브리지 회로(14)는 PCI 버스(6) 및 ISA 버스(15)를 접속한다. ISA 버스(15)는 입력 장치 제어기(16), 플로피 디스크 제어기(17), 평행 제어기(18) 및 RS232C 제어기(19)에 접속된다.

입력 장치 제어기(18)는 키보드 또는 마우스와 같은 입력 장치(20)에 접속된다. 플로피 디스크 제어기(17)는 플로피 디스크 드라이브(21)에 접속된다. 프린터 등은 평행 제어기(18)에 접속될 수 있다. 모뎀 등은 RS232C 제어기(19)에 접속될 수 있다.

초기 상태에서, ROM(2)에 저장된 부트 스트랩 프로그램이 개시되어 초기 설정을 수행한다. 그 후, 저장 장치(11 또는 12)가 액세스된다. 저장 장치(11 또는 12)에 저장된 오퍼레이팅 시스템이 판독된다. 오퍼레이팅 시스템은 주 메모리인 RAM(3)에 거주한다. 이에 의해, 오퍼레이팅 시스템이 개시된다. 오퍼레이팅 시스템의 제어 하에, 다양한 처리가 실행된다.

이 예에서는, PCI 버스 및 ISA 버스가 사용된다. 그러나, 본 발명에 따르면, USB(Universal Serial Bus)가 사용될 수 있다. 키보드, 마우스 등은 USB에 접속될 수 있다.

도 4에 도시된 데이터 처리 장치가 MPEG2 엔코딩 처리를 수행하는 경우, MPEG2 엔코딩 처리를 수행하기 위한 응용 프로그

램이 실행된다. 응용 프로그램은 IDE 하드 디스크와 같은 저장 장치(11) 또는 SCSI 하드 디스크와 같은 저장 장치(12)에 실행가능한 프로그램으로서 저장된다. 응용 프로그램이 실행되면, RAM(3)에 판독되고, CPU(1)에 의해 순차적으로 실행된다.

MPEG2 엔코딩 처리를 수행하기 위한 응용 프로그램은 IDE 하드 디스크와 같은 저장 장치(11) 또는 SCSI 하드 디스크와 같은 저장 장치(12)에 미리 인스톨될 수 있다. 또 다른 방식으로서, CD-ROM 또는 플로피 디스크에 실행가능 포맷 또는 압축 포맷으로 MPEG2 엔코딩 처리를 수행하기 위한 응용 프로그램을 저장할 수 있다. 사용자는 IDE 하드 디스크와 같은 저장 장치(11) 또는 SCSI 하드 디스크와 같은 저장 장치(12)에 저장된 프로그램을 인스톨할 수 있다. 또 다른 방식으로서, 통신 회선을 통해 응용 프로그램을 다운로드할 수 있다.

MPEG2 엔코딩 처리를 수행하기 위한 응용 프로그램이 실행되는 경우, 움직임 벡터 산출 처리, DCT 산출 처리, 양자화 처리 및 가변 길이 코드 엔코딩 처리가 예측 모드에 대응하는 디지털 비디오 데이터에 대해 수행된다. 디지털 비디오 데이터는 MPEG2 방법에 따라 압축된다. 이 때, 작업 영역으로서, RAM(3)이 사용된다. 이러한 처리를 위한 산출 동작은 CPU(1)의 함수들을 산출함으로써 수행된다. 디지털 비디오 데이터는 IEEE 1394 제어기(10)에 접속된 외부 디지털 VCR 등으로부터 입력된다. 출력 데이터는 SCSI 제어기(8) 또는 IDE 제어기(7)에 접속된 하드 디스크 드라이브 등에 기록된다.

도 5는 프로그램의 MPEG2 엔코딩 처리를 도시한 순서도이다.

도 5에 도시한 바와 같이, 복수의 프레임의 디지털 비디오 데이터가 입력된다. 디지털 비디오 데이터는 RAM(3)에 버퍼링된다 (단계 S1). 블록 매칭 처리에 의해, 움직임 벡터가 산출된다 (단계 S2). 블록 매칭 처리에서, 블록의 윤곽 픽셀(contour pixel)이 사용될 수 있다.

예측 모드가 I 화상, P 화상, 또는 B 화상인지의 여부가 판단된다 (단계 S3). 단계 S3에서의 판단 결과, 예측 모드가 I 화상이면, 동일 프레임의 (8×8) 픽셀의 각각의 블록에 대한 DCT 처리가 수행된다 (단계 S4). 구해진 계수 데이터가 양자화되고 (단계 S5), 이어서 가변 길이 코드로 엔코딩된다 (단계 S6). 최종 데이터는 참조 화상의 데이터로서 RAM(3)에 저장된다 (단계 S7).

단계 S3에서의 판단 결과, 예측 모드가 P 화상이면, 순방향 참조 화상(forward reference pixel)의 데이터가 RAM(3)으로부터 판독된다 (단계 S8). 참조 화상은 단계 S2에서 산출된 움직임 벡터에 대응하여 움직임 보상된다 (단계 S9). 이에 의해, 현재 화상의 데이터와 움직임 보상된 참조 화상의 데이터 간의 차이가 구해진다. 현재 화상의 데이터와 참조 화상의 데이터 간의 차이에 대해 DCT 처리가 수행된다 (단계 S10). 얻어진 데이터는 양자화되며 (단계 S11), 가변 길이 코드로 엔코딩된다 (단계 S12). 최종 데이터는 참조 화상의 데이터로서 RAM(3)에 저장된다 (단계 S13).

단계 S3에서의 판단 결과, 예측 모드가 B 화상이면, 양방향 참조 화상의 데이터가 RAM(3)으로부터 판독된다 (단계 S14). 단계 S2에서 산출된 움직임 벡터에 대응하여 참조 화상이 움직임 보상된다 (단계 S15). 현재 화상의 데이터와 움직임 보상된 참조 화상의 데이터 간의 차이가 구해진다. 현재 화상의 데이터와 참조 화상의 데이터 간의 차이에 대해 DCT 처리가 수행된다 (단계 S16). 구해진 데이터는 양자화되고 (단계 S17), 가변 길이 코드로 엔코딩된다 (단계 S18).

도 5에 도시한 단계 S2에서 산출되는 움직임 벡터는 다음의 방식으로 수행된다. 처리될 현 프레임으로부터 분할된 블록과 동일한 크기 및 동일한 원점을 가진 블록이 참조 프레임으로부터 추출된다. 참조 프레임의 블록이 선정된 검색 영역에서 이동되는 동안, 참조 프레임의 블록의 픽셀과 현 프레임의 관련 블록의 픽셀간의 차분값의 절대값들의 합이 잔여분으로서 구해진다. 최소의 잔여분을 갖는 참조 프레임의 블록이 구해진다. 블록 매칭 처리는 많은 산출 단계들을 요한다.

이로써, 본 발명에 따르면, 블록들의 데이터를 직교 변환하여 블록들을 비교함으로써 블록 매칭 처리가 수행된다. 직교 변환 처리의 예로서, 하다마드(Hadamard) 변환 방법이 사용된다.

즉, 도 6에 도시한 바와 같이, 현 프레임의 (16×16) 픽셀의 블록(CBLK)의 데이터(CD1, CD2, …, 및 CD256)가 구해진다. 도 7에 도시한 바와 같이, 현 프레임의 (16×16) 픽셀의 블록(CBLK)은 네 개의 블록(TBLK_C1 내지 TBLK_C4)으로 분할되며

, 이들 각각은 (8×8) 픽셀로 이루어진다. 도 8에 도시한 바와 같이, 네 개의 블록(TBLK_C1 내지 TBLK_C4)은 스펙트럼 데이터(TCD1-1 내지 TCD1-64, TCD2-1 내지 TCD2-64, TCD3-1 내지 TCD3-64 및 TCD4-1 내지 TCD4-64)로 직교 변환된다. 네 개의 블록(TBLK_C1 내지 TBLK_C4)의 데이터는 도 9에 도시한 바와 같은 지그재그 스캐닝 방법에 의해 낮은 공간 주파수 데이터의 순서로 구해진다.

마찬가지로, 도 10에 도시한 바와 같이, 참조 프레임의 (16×16) 픽셀의 블록(RBLK)의 데이터(RD1, RD2, …, 및 RD256)가 구해진다. 도 11에 도시한 바와 같이, 참조 프레임의 블록(RBLK)은 네 개의 블록(TBLK_R1 내지 TBLK_R4)으로 분할된다. 도 12에 도시한 바와 같이, 네 개의 블록(TBLK_R1 내지 TBLK_R4)은 스펙트럼 데이터(TRD1_1 내지 TRD1-64, TRD2_1 내지 TRD2-64, TRD3_1 내지 TRD3-64, 및 TRD4_1 내지 TRD4-64)로 직교 변환된다. 네 개의 블록(TBLK_R1 내지 TBLK_R4)의 데이터는 도 9에 도시한 바와 같이 지그재그 스캐닝 방법에 의해 낮은 공간 주파수 데이터의 순서로 구해진다.

비디오 신호가 직교 변환되는 경우, 에너지가 낮은 주파수 데이터 상에 집중된다. 이에 의해, 고 주파수 데이터는 거의 존재하지 않는다. 따라서, 네 개의 블록(TBLK_C1 내지 TBLK_C4)의 데이터가 지그재그 스캐닝 방법에 의해 구해지는 경우, 데이터의 수는 선정된 값으로 제한된다. 이 예에서는, 구해진 데이터의 수가 10으로 제한된다. 그러나, 본 발명에 따르면, 구해진 데이터의 수는 10 이외의 값일 수도 있다. 마찬가지로, 참조 프레임의 네 개의 블록(TBLK_R1 내지 TBLK_R4)의 데이터가 지그재그 스캐닝 방법에 의해 구해지는 경우, 구해진 데이터의 수는 선정된 값으로 제한된다. 이 예에서는, 구해진 데이터의 수가 10으로 제한된다.

즉, 예를 들어, 현 프레임의 네 개의 블록(TBLK_C1 내지 TBLK_C4)으로부터 10 개의 데이터 (도 8에서 흑점으로 표시됨)가 구해진다. 마찬가지로, 예를 들어, 참조 프레임의 네 개의 블록(TBLK_R1 내지 TBLK_R4)으로부터 10 개의 데이터 (도 12에서 흑점으로 표시됨)가 구해진다. 현 프레임의 네 개의 블록(TBLK_C1 내지 TBLK_C4)으로부터 구해진 데이터 및 참조 프레임의 네 개의 블록(TBLK_R1 내지 TBLK_R4)으로부터 구해진 데이터 간의 차분값의 절대값의 합이 잔여분으로서 구해진다.

블록 매칭 처리에서, 하나의 블록의 데이터는 직교 변환되고 데이터의 수는 선정된 값으로 제한되므로, 산출 단계의 수는 현저히 감소될 수 있다. 이에 의해, 산출 속도가 향상된다.

즉, 상술한 바와 같이, 하나의 블록은 네 개의 블록으로 분할된다. 네 개의 블록은 직교 변환된다 (예를 들어 하다마드 변환 방법에 의해). 각각 직교 변환된 블록의 데이터의 수는 10으로 제한된다. 이 상태에서, 블록 매칭 처리가 수행된다. 이 경우, 데이터의 수는 10으로 제한되고 하나의 블록은 네 개의 블록으로 분할되므로, 블록 매칭 처리에서 잔여분을 구하기 위한 산출 단계의 수는 40이다. 이에 반해, 블록 매칭 처리가 (16×16) 픽셀로 이루어진 블록으로 수행되는 경우, 산출 단계의 수는 (16×16 = 256)이 된다. 이로써, 직교 변환되는 하나의 블록으로 잔여분을 구하는 경우, 산출 단계의 수를 현저히 줄일 수 있다.

이 경우, 하다마드 변환 방법과 같은 직교 변환 방법이 사용된다. 그러나, 하다마드 변환 방법은 가산 및 감산과 같은 간단한 산술로 수행될 수 있다. 이에 의해, 산출 단계의 수는 크게 증가하지 않는다.

MPEG2 엔코딩 처리에서는, 현 프레임의 화상이 다음 참조 프레임의 화상으로서 사용된다. 이에 의해, 현 프레임의 화상의 블록의 직교 변환된 데이터가 저장되는 경우, 참조 프레임의 데이터로서 사용될 수 있다.

움직임 벡터가 검색되는 경우, 검색 영역들이 중첩된다. 중첩된 영역에서는, 동일한 직교 변환 데이터가 요구된다. 이에 의해, 참조 프레임의 블록에 대하여, 픽셀마다 이동된 직교 변환 데이터가 저장된다. 이 경우, 검색 영역들이 중첩될 때, 저장된 데이터가 사용될 수 있다.

상술한 예에서, 직교 변환 방법으로서, 하다마드 변환 방법이 사용되었다. 그러나, 본 발명에 따르면, 예를 들어, DCT 변환 방법 또는 FFT(Fast Fourier Transform)가 사용될 수 있다.

상술한 예에서, (16×16) 픽셀의 블록은 각각이 (8×8) 픽셀로 이루어지는 네 개의 블록들로 분할된다. 네 개의 분할된 블록들은 직교 변환된다. 또 다른 방식으로서, (16×16) 픽셀의 블록은 직접 직교 변환될 수 있다. 그러나, 하나의 블

록이 네 개의 블록으로 분할되고 네 개의 블록이 직교 변환되는 경우, 변환 알고리즘이 간단해진다. 아울러, 범용 직교 변환 회로 및 알고리즘이 사용될 수 있다.

다음으로, 본 발명의 제2 실시예를 설명한다. 제2 실시예에 따르면, 블록 매칭 산출 루프의 중간에, 현 프레임의 블록의 픽셀들과 참조 프레임의 관련 블록의 픽셀들 간의 차분값의 절대값의 합이 선정된 임계값과 비교된다. 이 합이 선정된 임계값 이상이면, 처리가 종료된다. 이로써, 산출 단계의 수를 줄일 수 있다. 따라서, 움직임 벡터가 고속으로 검출될 수 있다.

임계값은 원점에서의 잔여분 AD(0, 0)와 평균 이산 절대값(MAD)의 합에 대응하여 할당된다. 이로써, 임계값이 동적으로 할당되므로, 처리가 효과적으로 수행될 수 있다.

P 화상 및 B 화상의 경우, 프레임 내 엔코딩 처리가 각각의 매크로 블록에 대해 수행될 수 있다 (도 5에서는, 편의상, P 화상 및 B 화상의 경우, 프레임 간 엔코딩 처리가 참조 프레임으로 수행됨). 즉, 프레임 간 엔코딩 처리는 프레임 내 엔코딩 처리보다 효과적으로 화상을 압축할 수 있다. 그러나, 많은 DC 성분을 포함하는 화상 또는 크게 이동하는 화상의 경우 (즉, 현 프레임의 블록의 픽셀들과 참조 프레임의 관련 블록의 픽셀들간의 차분값의 절대값의 합이 큰 경우), 프레임 내 엔코딩 처리는 프레임 간 엔코딩 처리보다 효과적으로 화상을 압축할 수 있다. 프레임 내 엔코딩 처리가 수행되는 경우, 움직임 벡터 산출 처리가 요구되지 않으므로, 움직임 벡터의 비정확성이 허용될 수 있다.

현재 블록 중 하나의 블록의 픽셀들과 참조 블록의 관련 블록의 픽셀들간의 차분값의 절대값의 합이 선정된 임계값 이상인 경우, 처리는 종료된다. 따라서, 만일 임계값에 큰 값이 할당되면, 처리가 중간에 종료될 확률은 높아진다. 이에 반해, 임계값에 작은 값이 할당되는 경우, 움직임 벡터가 비정확하게 검출될 확률이 높아진다. 그러나, 프레임 내 엔코딩 처리가 수행되는 경우, 움직임 벡터 산출 처리가 요구되지 않으므로, 움직임 벡터의 비정확성이 허용될 수 있다. 이로써, 프레임 내 엔코딩 처리에서 얻어진 잔여분이 임계값으로서 사용되는 경우, 처리의 효율성이 향상된다.

P 화상 및 B 화상이 엔코딩되는 경우, 프레임 내 엔코딩 처리는 검출된 움직임 벡터의 잔여분 AD(x, y) 및 MAD 값에 대응하여 수행된다.

MAD는 하나의 프레임의 픽셀들의 값 및 그 평균값간의 차분값의 절대값의 합이다. MAD는 다음과 같이 구해진다.

$$MAD = \sum_{x=1}^{16} \sum_{y=1}^{16} |Pixel(x,y) - AVE\_BLOCK|$$

$$AVE\_BLOCK = \frac{\sum_{x=1}^{16} \sum_{y=1}^{16} Pixel(x,y)}{256}$$

MAD는 화상의 한 블록의 패턴의 복잡성을 나타낸다. 따라서, 패턴이 단순하면, MAD 값은 작게 된다. 반대로, 패턴이 복잡하면, MAD 값은 커진다.

따라서, 도 13에 도시된 함수에 의해, 프레임 간 엔코딩 처리 또는 프레임 내 엔코딩 처리가 수행되는지의 여부가 결정된다. 도 13에 있어서, 수평축은 움직임 벡터의 위치에서 잔여분 AD(x, y)의 값을 나타낸다. 반면에, 수직축은 MAD 값을 나타낸다. 도 13에 있어서, MAD 값과 움직임 벡터의 위치에서의 잔여분 (x, y)의 값 양자 모두가 영역 AR1에 있는 경우에는, 프레임 간 엔코딩 처리가 수행된다. 이들이 영역 AR2에 있는 경우에는, 프레임 내 엔코딩 처리가 수행된다. MAD 값이 작으면, 현 블록의 패턴이 단순하기 때문에, 이 함수는 프레임 내 엔코딩 처리가 수행됨을 나타낸다. 움직임 벡터의 위치에서의 잔여분 AD(x, y)의 값이 작으면, 이런 함수는, 프레임 간 엔코딩 처리가 프레임 내 엔코딩 처리 대신에 수행됨을 나타낸다.

블록 매칭 처리 중간에, 현재 프레임의 블록의 픽셀들과 참조 프레임의 관련 블록의 픽셀들간의 차분값의 절대값의 합이, 선정된 임계값과 비교된다. 합계가 임계값 이상이면, 처리는 종료된다. 또한, 원점에서의 잔여분 AD(0, 0)의 값 및 MAD 값을 이용하여 도 13 및 14의 함수에 따라, 프레임 간 엔코딩 처리 또는 프레임 내 엔코딩 처리가 수행되는지의 여부가 결정된다. 또한, 움직임 보상이 수행되는지의 여부가 결정된다. 결정 결과에 따라, 임계값의 초기값이 할당된다. 따라서, 움직임 벡터는 효율적으로 산출될 수 있다. 도 15는 이런 처리를 도시한 순서도가다.

블록 매칭 처리에서 초기에 할당된 임계값이 항상 작은 것은 아니므로, 움직임 벡터가 처음으로 검출되면, 임계값은 MAD 값 및 원점에서의 잔여분 AD(0, 0)의 값으로 얻어진다. 동일 블록에 대한 다음 블록 매칭 처리에 있어서, 본래의 임계값 또는 얻어진 합 중 더 작은 것이면 어느 것이나 사용된다. 따라서, 처리는 효율적으로 수행될 수 있다.

도 15에 있어서, 참조 프레임의 블록의 검색 영역이 초기에 설정된다(단계 S21). 그 후, MAD 값이 얻어진다(단계 S22). 원점에서 잔여분 AD(0, 0)의 값이 얻어진다(단계 S23). MAD 값 및 원점에서의 잔여분 AD(0, 0)의 값에 대응하여, ADmin의 초기값이 설정된다(단계 S24).

ADmin의 값은 얻어진 잔여분의 최소값을 나타낸다. ADmin의 초기값은 현 프레임의 블록의 픽셀들과 참조 프레임의 관련 블록의 픽셀들 간의 차분값의 절대값의 합의 임계값의 초기값이 된다. ADmin의 값은, 프레임 간 엔코딩 처리 또는 프레임 내 엔코딩 처리가 수행되는지의 여부 또는 움직임 보상 처리가 도 13 및 14에 도시된 함수에 따라 수행되는지의 여부에 따라 동적으로 설정된다.

단계 S24에서 ADmin의 초기값이 설정되면, 검색 영역의 상부 우측 위치는 제1 블록 매칭 처리에 대해 선택된다(단계 S25). 초기 위치에서 한 블록에 대한 블록 매칭 처리가 수행된다(단계 S26).

블록 매칭 처리의 중간에, 현재 프레임의 블록의 픽셀들과 참조 프레임의 관련 블록의 픽셀들 간의 차분값의 절대값의 합이 선정된 임계값과 비교된다. 합계가 선정된 임계값 이상이면, 처리는 종료된다. 블록 매칭 처리가 초기에 수행되면, 단계 S24에서 임계값으로서 얻어진 ADmin의 초기값이 사용된다. 도 16은 블록 매칭 처리를 도시한 순서도가다.

도 16을 참조하면, 블록 매칭 처리(단계 S26)에 있어서, 픽셀 위치는 초기에 설정된다(단계 S41). 잔여분 AD의 값은 초기에 설정된다(단계 S42). 잔여분은 현재 프레임의 픽셀들과 참조 프레임의 픽셀들 간의 차분값의 절대값의 합으로 얻어진다(단계 S43). 블록 매칭 처리의 중간에서, 참조 프레임의 픽셀들과 현재 프레임의 픽셀들 간의 차분값의 절대값의 합이 ADmin의 값을 초과하는지의 여부가 결정된다(단계 S44). 단계 S44에서 결정된 결과가 '예' 이면 (즉, 합계가 ADmin을 초과함), 블록 매칭 처리는 종료된다. 흐름은 주 루틴으로 복귀한다. 단계 S44에서 결정된 결과가 '아니오' 이면 (즉, 합계가 ADmin의 값을 초과하지 않음), 블록 매칭 처리가 모든 픽셀들에 대해 수행되는지의 여부가 결정된다(단계 S45). 단계 S45에서 결정된 결과가 '아니오' 이면 (즉, 블록 매칭 처리가 모든 픽셀들에 대해 수행되지 않음), 흐름은 단계 S43으로 복귀한다. 단계 S43에서, 참조 프레임의 픽셀들과 현재 프레임의 픽셀들 간의 차분값의 절대값의 합이 얻어진다. 단계 S45에서 결정된 결과가 '예' 이면 (즉, 블록 매칭 처리가 모든 픽셀들에 대해 수행됨), 블록 매칭 처리는 종료된다. 그 후, 흐름은 주 루틴으로 복귀한다.

상술한 바와 같이, 블록 매칭 처리에 있어서, 단계 S44에서 AD 값이 ADmin 값을 초과하는지의 여부가 결정된다. 단계 S44에서 결정된 결과가 '예' 이면 (즉, AD 값이 ADmin 값을 초과함), 흐름은 주 루틴으로 복귀한다. 따라서, ADmin 값이 임계값이 된다. 블록 매칭 처리의 중간에서, 현 프레임의 블록의 픽셀들과 참조 프레임의 관련 블록의 픽셀들 간의 차분값의 절대값의 합이 임계값과 비교된다. 합계가 선정된 임계값을 초과하면, 블록 매칭 처리는 종료된다. 따라서, 산출 단계의 수가 감소된다. 결과적으로, 움직임 벡터가 고속으로 검출될 수 있다.

또한, 임계값의 초기값으로서 사용되는 ADmin 값은, MAD 값 및 원점에서의 잔여분 AD (0, 0)의 값에 따라 설정된다. 잔여분이, 프레임 간 엔코딩 처리가 수행됨을 나타내면, 움직임 벡터가 요구되지 않기 때문에, 움직임 벡터의 부정확성이 허용가능하게 된다. 임계값이 동적으로 변화하기 때문에, 잔여분이 움직임 벡터에 요구되지 않는 값을 초과하면, 블록 매칭 처리가 종료될 가능성이 높아진다. 따라서, 산출 단계의 수는 더욱 감소된다.

도 15에 있어서, 블록 매칭 처리에서 얻어진 AD 값은 얻어진 최소값 ADmin과 비교된다(단계 S27). 단계 S27에서 결정된

결과가 '예' 이면 (즉, AD 값이 최소값 ADmin 보다 작음), 현재의 합계 AD가 최소값 ADmin으로써 사용된다(단계 S28). AD 값이 움직임 벡터 MV로서 기록된다(단계 S29). 그 후, 다음 블록이 처리된다(단계 S30). 그 후, 최종 블록이 처리되었는지의 여부가 결정된다(단계 S31). 단계 S31에서 결정된 결과가 '아니오' 이면 (즉, 최종 블록이 처리되지 않음), 흐름은 단계 S26으로 복귀한다. 단계 S26에서, 블록 매칭 처리는 다음 블록에 대해 수행된다.

도 16에 도시된 바와 같이, 블록 매칭 처리의 중간에서, 현재 프레임의 블록의 픽셀들과 기준 블록의 관련 블록의 픽셀들 간의 차분값의 절대값의 합이 선정된 임계값과 비교된다. 합이 선정된 임계값 이상이면, 처리는 종료된다. 임계값으로서, ADmin 값이 사용된다.

단계 S27에서, 블록 매칭 처리에서 얻어진 AD 값이 얻어진 ADmin 값과 비교된다. 단계 S27에서 결정된 결과가 '예' 이면 (즉, AD 값이 현재 최소값 ADmin 보다 작음), AD 값은 ADmin 값이 된다. 따라서, 얻어진 AD 값이 ADmin 값보다 작으면, 다음 임계값은 AD의 최소값이 된다. 따라서, 도 16에 도시된 블록 매칭 처리에 있어서, 잔여분이 ADmin 값을 초과하면, 처리는 종료된다.

그 후, 단계 S26에서 단계 S31까지의 루프가 반복된다. 현재 프레임의 블록의 픽셀들과 참조 프레임의 관련 블록의 픽셀들 간의 차분값의 절대값의 합의 최소값이 얻어진다. 단계 S31에서, 최종 블록이 처리되었는지의 여부가 결정된다. 단계 S31에서 결정된 결과가 '예' 이면 (즉, 최종 블록이 처리됨), 최소값 ADmin은 움직임 벡터 MV가 된다. 그 결과가 저장된다(단계 S32).

상술한 예에 있어서, 임계값의 초기값은 MAD 값 및 원점에서의 잔여분 AD(x, y) 값과 대응하여 할당된다. 그러나, 본 발명에 따르면, 임계값은 MAD 값 및 원점에서의 잔여분 AD (x, y) 값중 하나에 대응하여 할당될 수 있다.

본 발명의 제3 실시예에 있어서, 도 17에 도시된 바와 같이, 블록 매칭 처리는 참조 프레임의 블록의 픽셀들 및 현재 프레임의 관련 블록의 픽셀들을 바둑판 모양으로 속아냄으로써 수행된다.

도 17을 참조하면, 참조 프레임의 블록(31)은 16×16 픽셀들로 이루어진다. 8×8 픽셀들은 블록(31)로부터 바둑판 모양으로 얻어진다. 더욱이, 현재 프레임의 블록(32)은 16×16 픽셀들로 이루어진다. 8×8 픽셀들은 블록(31)로부터 바둑판 모양으로 얻어진다.

이점에서, 바둑판 모양으로 얻어지는 현재 프레임의 픽셀들 및 참조 프레임의 픽셀들은 블록 매칭 처리가 MMX 지시로 효율적으로 수행될 수 있도록 메모리(RAM(3)의 선정된 영역)에 연속 데이터로써 저장된다.

즉, 도 18a에 도시된 바와 같이, 현재 프레임의 픽셀들 및 참조 프레임의 픽셀들은 바둑판 모양으로 얻어진다. 도 18b에 도시된 바와 같이, 바둑판 모양으로 속아낸 픽셀들은 연속 데이터로써 재배열된다. 도 18c에 있어서, 바둑판 모양으로 속아낸 픽셀들은 메모리의 연속 어드레스에 저장된다.

현재 프레임의 픽셀들 및 참조 프레임의 픽셀들이 메모리에 연속 데이터로써 저장될때, 블록 매칭 처리가 MMX 지시로 수행되기 때문에, 처리는 고속도로 수행될 수 있다.

바둑판 모양으로 속아낸 참조 프레임의 픽셀들 및 현재 프레임의 픽셀들이 메모리에 연속 데이터로써 저장되면, 두 픽셀들이 동시에 검색되기 때문에, 대수 검색 처리가 용이하게 수행될 수 있으며, 또한 MMX 명령도 이용될 수 있다.

대수 처리에 있어서, 최소 잔여분을 갖는 지점이 검색 영역에서 대략 검색된다. 그 후, 최소 잔여분을 갖는 지점은 대략 검색된 영역 주위를 미세 검색한다. 그 결과, 움직임 벡터가 검출된다.

현재 프레임의 픽셀들 및 구해진 참조 프레임의 픽셀들이 메모리에 연속 데이터로써 저장될때, 대수 검색 처리는 다음의 방식으로 수행된다.

현재 프레임의 모든 픽셀들 및 참조 프레임의 모든 픽셀들을 저장하는 제1 메모리, 및 연속 데이터로써 바둑판 모양으로 얻어진 참조 프레임의 픽셀들 및 현재 프레임의 픽셀들을 저장하는 제2 메모리 (또는 메모리 영역)가 제공된다. 제2 메

모리를 사용하여, 움직임 벡터는 2 픽셀 스텝에서 대략 검색된다. 움직임 벡터가 대략 검색된 후, 제1 메모리를 사용하여 움직임 벡터는 픽셀별로 얻어진 포인트 근처에서 미세하게 검색된다. 따라서, 움직임 벡터는 정밀하게 검출될 수 있다.

예를 들어, 도 19a에 도시된 바와 같이, 각각의 프레임의 화상 데이터편 F1, F2, F3, F4, F5, F6, F7 등이 입력된다. 입력된 화상 데이터편 F1, F2, F3, F4, F5, F6, F7 등은 I, B, B, P, B, B 및 P 화상순으로 MPEG2 화상 데이터편 P1,P2, P3, P4, P5, P6, P7 등으로 엔코딩된다.

이러한 엔코딩 처리에서는 도 20에 도시된 바와 같이, 움직임 벡터를 얻기 위하여, 작업 영역(RAM 3)은 메모리 영역(21A 내지 21F) 및 메모리 영역(22A 내지 22C)를 가진다. 메모리 영역(21A 내지 21F)는 한 프레임의 모든 픽셀들을 저장한다. 메모리 영역(22A 내지 22C)는 연속 데이터로써 바둑판 모양으로 얻어진 한 프레임의 픽셀들을 저장한다. 화상 데이터편 F1, F2, F3 등이 도 21에 도시된 바와 같이 입력되면, 각각의 프레임의 화상 데이터편은 메모리 영역(21A 내지 21F)에 저장된다. 또한, 픽셀들은 샘플별로 화상 데이터편으로부터 바둑판 모양으로 얻어진다. 최종 픽셀들은 연속 어드레스에 배열되고 화상 데이터편 F1, F2, F3 등으로써 메모리 영역(22A 내지 22C)에 저장된다.

즉, 시점 T1에서, 화상 데이터편 F1은 메모리 영역(21A)에 저장된다. 시점 T2에서, 화상 데이터편 F2는 메모리 영역(21B)에 저장된다. 시점 T3에서, 화상 데이터편 F3는 메모리 영역(21C) 에 저장된다. 시점 T4에서, 화상 데이터편 F4는 메모리 영역(21D)에 저장된다.

시점 T4에서, 픽셀들은 샘플별로 화상 데이터편으로부터 바둑판 모양으로 얻어진다. 연속 어드레스로 배열된 화상 데이터편 F1은 메모리 영역(22A)에 저장된다. 픽셀들은 샘플별로 화상 데이터편 F4로부터 바둑판 모양으로 얻어진다. 연속 어드레스로 배열된 화상 데이터편 F4는 메모리 영역(22B)에 저장된다.

시점 T5에서, 화상 데이터편 F5는 메모리 영역(21F)에 저장된다. 픽셀들은 샘플별로 화상 데이터편으로부터 바둑판 모양으로 얻어진다. 연속 어드레스로 배열된 화상 데이터편은 메모리 영역(22C)에 저장된다.

시점 T6에서, 화상 데이터편 F6는 메모리 영역(21F)에 저장된다. 픽셀들은 샘플별로 화상 데이터편 F3로부터 바둑판 모양으로 얻어진다. 연속 어드레스로 배열된 화상 데이터편 F3는 메모리 영역(22C)에 저장된다.

시점 T7에서, 화상 데이터편 F7은 메모리 영역(21A)에 저장된다. 픽셀들은 샘플별로 화상 데이터편으로부터 바둑판 모양으로 얻어진다. 연속 어드레스로 배열된 화상 데이터편 F7은 메모리 영역(22A)에 저장된다.

도 21에 도시된 바와 같이, 각각의 프레임의 화상 데이터편은 메모리 영역(21A 내지 21F)에 저장된다. 또한 픽셀들은 샘플별로 화상 데이터편으로부터 바둑판 모양으로 얻어진다. 연속 어드레스로 배열된 화상 데이터편은 메모리 영역(22A 내지 22C)에 저장된다.

메모리 영역(21A 내지 21F)에 저장된 화상 데이터편 F1, F2, F3 등 및 메모리 영역(22A 내지 22C)에 저장된 화상 데이터편 F1, F2, F3 등에 의해, 움직임 벡터는 얻어진다. 움직임 벡터는 2 픽셀 스텝으로 선정된 검색 영역에서 검색된다. 움직임 벡터는 픽셀별로 검색된 포인트 근처에서 검색된다. 즉, 움직임 벡터는 대수 검색 처리에 의해 얻어진다.

화상 데이터편 P1은 I 화상이므로, 시점 T1에서 시점 T3까지 엔코딩될 수 있다.

시점 T4에서, P 화상인 화상 데이터편 P4가 엔코딩된다. 화상 데이터편 P4의 움직임 벡터가 얻어진다. 화상 데이터편 P4에 대해서, 화상 데이터편 F1은 참조 프레임으로서 사용되고, 화상 데이터편 F4는 현재 프레임으로서 사용된다. 이 경우, 검색 처리 중 (2 픽셀 스텝) 에서, 참조 프레임의 블록으로서 메모리 영역(22A)에 저장된 화상 데이터편 F1이 사용된다. 현재 프레임의 블록으로서, 메모리 영역(22B)에 저장된 화상 데이터편 F4가 사용된다. 미세 검색 처리중(한 픽셀 단계)에서, 참조 프레임의 블록으로서 메모리 영역(21A)에 저장된 화상 데이터편 F1이 사용된다. 현재 프레임의 블록으로서 메모리 영역(21D)에 저장된 화상 데이터편 F4가 사용된다.

시점 T5에서, B 화상인 화상 데이터편 P2가 엔코딩된다. 화상 데이터편 P2의 움직임 벡터가 얻어진다. 화상 데이터편 P2에 대하여, 참조 프레임으로서 화상 데이터편 F1 및 F4가 사용된다. 현재 프레임으로서, 화상 데이터편 F2가 사용된다. 이 경우, 대략 검색 처리(2 픽셀 스텝)에 있어서, 참조 프레임의 블록으로서 메모리 영역(22A)에 저장된 화상 데이터편 F1 및 메모리 영역(22B)에 저장된 화상 데이터편 F4가 사용된다. 현재 프레임의 블록으로서, 메모리 영역(22C)에 저장된 화상 데이터편 F2가 사용된다. 미세 검색 처리(1 픽셀 스텝)에 있어서, 참조 프레임의 블록으로서, 메모리 영역(21A)에 저장된 화상 데이터편 F1 및 메모리 영역(21D)에 저장된 화상 데이터편 F4가 사용된다. 현재 프레임의 블록으로서 메모리 영역(21B)에 저장된 화상 데이터편 F2가 사용된다.

시점 T6에서, B 화상인 화상 데이터편 P3가 엔코딩된다. 화상 P3의 움직임 벡터가 얻어진다. 화상 데이터편 P3에 있어서, 참조 프레임으로서, 화상 데이터편 F1 및 F4가 사용된다. 현재 프레임으로서, 화상 데이터편 F3가 사용된다. 이 경우, 대략적인 검색 처리(2 픽셀 스텝)에 있어서, 참조 프레임의 블록으로서, 메모리 영역(22A)에 저장된 화상 데이터편 F1 및 메모리 영역(22B)에 저장된 화상 데이터편 F4가 사용된다. 현재 프레임으로서, 메모리 영역(22C)에 저장된 화상 데이터편 F3가 사용된다. 미세 검색 처리(한 픽셀 단계)에 있어서, 참조 프레임의 블록으로서, 메모리 영역(21A)에 저장된 화상 데이터편 F1 및 메모리 영역(21D)에 저장된 화상 데이터편 F4가 사용된다. 현재 프레임의 블록으로서 메모리 영역(21C)에 저장된 화상 데이터편 F3가 사용된다.

시점 T7에서, P 화상인 화상 데이터편 P7이 엔코딩된다. 화상 P7의 움직임 벡터가 얻어진다. 화상 데이터편 P7에 대하여, 참조 프레임으로서 화상 데이터편 F4가 사용된다. 현재 프레임으로서, 화상 데이터편 F7이 사용된다. 이 경우, 대략 검색 처리(2 픽셀 스텝)에 있어서, 참조 프레임의 블록으로서, 메모리 영역(22B)에 저장된 화상 데이터편 F4가 사용된다. 현재 프레임의 블록으로서, 메모리 영역(22A)에 저장된 화상 데이터편 F7이 사용된다. 미세 검색 처리(한 픽셀 단계)에 있어서, 참조 프레임의 블록으로서, 메모리 영역(21D)에 저장된 화상 데이터편 F4가 사용된다. 현재 프레임의 블록으로서, 메모리 영역(21A)에 저장된 화상 데이터편 F7이 사용된다.

그 다음, 마찬가지로 시점 T8에서 B 화상인 화상 데이터편 P5의 움직임 벡터가 얻어진다. 시점 T9에서, B 화상인 화상 데이터편 P6의 움직임 벡터가 얻어진다.

도 22는 움직임 벡터를 산출하는 대수 검색 처리의 순서도를 도시한다. 도 22에 있어서, 입력 화상 데이터는 저장된다. 픽셀은 샘플별로 참조 프레임 및 현재 프레임으로부터 바둑판 모양으로 추출된다. 바둑판 모양으로 속아낸 픽셀들은 연속 데이터로서 배열되고 저장된다(단계 S121).

그 후, 화상의 모든 블록이 처리되었는지가 판정된다(단계 S122에서).

단계 S122에서 판정된 결과가 '아니오'인 경우(즉, 화상의 모든 블록이 처리되지 않은 경우), 2 픽셀 스텝에서 블록이 선정된 검색 영역으로 이동되고 있는 동안, 움직임 벡터가 검색된다(단계 S123).

움직임 벡터가 검출된 후, 검출된 움직임 벡터 픽셀 부근에서 블록이 픽셀 단위로 이동되면서, 움직임 벡터가 검색된다(단계 S124).

검출된 결과가 저장된다(단계 S125). 그 후, 다음 블록이 처리된다(단계 S126). 그 후, 단계 S122로 복귀한다. 단계 S122에서 판정된 결과가 '아니오'인 경우(즉, 모든 블록이 처리되지 않은 경우), 마찬가지의 처리가 반복된다. 따라서, 다음 블록의 움직임 벡터가 구해진다. 화상의 최종 블록의 움직임 벡터가 구해진 후, 단계 S122에서 판정된 결과가 '예'이기 때문에, 처리는 완료된다.

도 23은 도 22에 나타낸 단계 S123에서의 대략 검색 처리(coarse searching process)를 나타낸 순서도이다. 대략 검색 처리에서(2 픽셀 스텝에서), 현 프레임의 픽셀들과 참조 프레임의 픽셀들이 바둑판 모양으로 추출된다. 추출된 픽셀들은 연속 데이터로서 메모리에 저장된다.

도 23에서, 검색 영역의 개시점이 설정된다(단계 S131). 수직 검색 개시 위치는 상단에 리셋된다(단계 S132). 수직 방향에서, 하단이 검출되었는지의 여부가 판정된다(단계 S133). 단계 S133에서 판정된 결과가 '아니오'인 경우(즉, 하단이

검출되지 않은 경우), 수평 위치가 좌단에 리셋된다(단계 S134).

그 후, 검색 영역의 우단이 검출되었는지의 여부가 판정된다(단계 S135). 단계 S135에서 판정된 결과가 '아니오'인 경우(즉, 검색 영역의 우단이 검출되지 않은 경우), 블록 매칭 처리가 (8×8) 픽셀들의 바둑판 모양의 블록에 대하여 수행되고 잔여분이 구해진다(단계 S136).

이 때, 잔여분 AD가 구해진 최소값 ADmin보다 작은지의 여부가 판정된다(단계 S137). 단계 S137에서 판정된 결과가 '예'인 경우(즉, 잔여분 AD가 최소값 ADmin보다 작은 경우), 잔여분 AD는 최소값 ADmin이다. 또한, 움직임 벡터 VT는 현재 위치이다(단계 S138). 그 후, 수평 위치가 2개의 픽셀만큼 이동된다(단계 S139). 현 프레임의 픽셀들과 참조 프레임의 픽셀들이 바둑판 모양으로 추출되고 연속 데이터로서 메모리에 저장된다. 따라서, 수평 위치는 2개의 픽셀에 대해 이동되고, 어드레스는 메모리 내의 하나의 위치에 대해 이동된다.

단계 S137에서 판정된 결과가 '아니오'인 경우(즉, 잔여분 AD가 최소값 ADmin보다 작지 않은 경우), 단계 S139로 진행한다. 단계 S139에서, 수평 위치가 2개의 픽셀에 대해 이동된다. 그 후, 단계 S135로 복귀한다.

단계 S135에서, 검색 영역의 우단이 검출되었는지의 여부가 판정된다. 단계 S135에서 판정된 결과가 '아니오'인 경우(즉, 검색 영역의 우단이 검출되지 않은 경우), 마찬가지의 처리가 반복된다. 따라서, 블록이 우단으로 이동되고 있는 동안, 잔여분이 구해진다. 최소 잔여분은 최소 ADmin으로서 저장된다.

단계 S135에서 판정된 결과가 '예'인 경우(즉, 검색 영역이 우단이 검출된 경우), 블록이 2개의 픽셀에 대해 수직으로 이동된다(단계 S140). 그 후, 단계 S133으로 복귀한다. 그 후, 마찬가지의 처리가 수행된다.

단계 S133에서 판정된 결과가 '예'인 경우(즉, 검색 영역의 하단이 검출된 경우), 그 결과가 움직임 벡터 MV로서 저장된다(단계 141). 움직임 벡터 MV는 미세 검색 처리(fine searching process)의 참조점으로 된다.

도 24는 도 22에 나타낸 단계 S24에서의 미세 검색 처리를 나타낸 순서도이다. 미세 검색 처리에서, 현 프레임의 모드 픽셀들과 참조 프레임의 모든 픽셀들을 저장하는 메모리가 사용된다.

도 24에서, 도 23에 나타낸 단계 S141에서 구해진 참조점의 좌상단에서 개시점이 설정된다(단계 S151). 수직 검색 개시 위치는 상단에 리셋된다(단계 S152). 그 후, 검색 영역의 하단이 검출되었는지의 여부가 판정된다(단계 S153). 단계 S153에서 판정된 결과가 '아니오'인 경우(즉, 하단이 검출되지 않은 경우), 수평 위치가 좌단에 리셋된다(단계 S154).

그 후, 검색 영역의 후단이 검출되었는지의 여부가 판정된다(단계 S155). 단계 S155에서 판정된 결과가 '아니오'인 경우(즉, 검색 영역의 우단이 검출되지 않은 경우), 블록 매칭 처리가 (16×16) 픽셀들에 대해 수행되고 잔여분이 구해진다(단계 S156).

그 후, 잔여분 AD가 구해진 최소값 ADmin보다 작은지의 여부가 판정된다(단계 S157). 단계 S157에서 판정된 결과가 '예'인 경우(즉, 잔여분 AD가 최소값 ADmin보다 작은 경우), 잔여분 AD는 최소값 ADmin이다(단계 S158). 움직임 벡터 MV는 현재 위치이다. 수평 위치는 하나의 픽셀에 대해 이동된다(단계 S159).

단계 S157에서 판정된 결과가 '아니오'인 경우(즉, 잔여분 AD가 최소값 ADmin보다 작지 않은 경우), 단계 S159로 진행한다. 단계 S159에서, 수평 위치가 하나의 픽셀에 대하여 이동된다. 그 후, 단계 S155로 복귀한다.

단계 S155에서, 검색 영역의 우단이 검출되었는지의 여부가 판정된다. 단계 S155에서 판정된 결과가 '아니오'인 경우(즉, 검색 영역의 우단이 검출되지 않은 경우), 마찬가지의 처리가 반복된다. 따라서, 블록이 검색 영역의 좌측에서 우측으로 이동되면서, 잔여분이 구해진다. 구해진 최소 잔여분이 최소값 ADmin으로서 저장된다.

단계 S155에서 판정된 결과가 '예'인 경우(즉, 검색 영역의 우단이 검출된 경우), 블록이 하나의 픽셀에 대하여 수직 방향으로 이동된다(단계 S160). 그 후, 단계 S153으로 복귀한다. 그 후, 마찬가지의 처리가 반복된다.

단계 S153에서 판정된 결과가 '예'인 경우(즉, 검색 영역의 하단이 검출된 경우), 움직임 벡터 VM이 구해지고 처리가 종료된다.

상술한 예에서, 참조 프레임의 픽셀들과 현 프레임의 픽셀들은 바둑판 모양으로 추출된다. 그러나, 본 발명에 따르면, 솎아내는(thin-out) 단계 및 솎아내는 방법은 상술한 예에 한정되지 않는다.

상술한 예에서, 대수 검색 처리가 수행된 경우, 대략 검색 처리에 의해 각 샘플에 대하여 솎아내는 픽셀들을 저장하는 메모리 내에서, 어드레스가 각각의 위치에 대하여 이동되어, 움직임 벡터가 2 픽셀 스텝으로 검색된다. 대안적으로, 2개의 위치에 대하여 어드레스를 동시에 이동시킴으로써, 4개의 픽셀에 대하여 검색 처리가 동시에 수행된다. 또한, 3개의 위치에 대하여 어드레스를 동시에 이동시킴으로써, 6개의 픽셀에 대한 검색 처리가 동시에 수행될 수 있다. 상술한 예에서, 대수 검색 처리는 2개의 픽셀에 대하여 동시에 행하는 대략 검색 처리 및 하나의 픽셀 스텝에서 행하는 미세 검색 처리에 의해 행해진다. 선택적으로, 대수 검색 처리는 복수의 단에서 행해질 수 있다.

본 발명에 따르면, 참조 픽셀들과 현 프레임의 픽셀들이 바둑판 모양으로 솎아낸 다음 블록 매칭 처리가 수행된다. 이때, 현 프레임의 픽셀들과 참조 프레임의 픽셀들이 연속 데이터로서 메모리에 저장된다. 따라서, 블록 매칭 처리가 수행된 때, MMX 명령이 효과적으로 사용될 수 있으므로, 고속으로 처리가 수행될 수 있다.

또한, 현 프레임의 픽셀들과 참조 프레임의 픽셀들을 저장하는 제1 메모리와, 바둑판 모양으로 솎아내는 현 프레임의 픽셀들과 참조 프레임의 픽셀들을 저장하는 제2 메모리가 준비된다. 제2 메모리에 대하여, 2 픽셀 스텝에서 대략 검색 처리가 수행된다. 이 경우, 바둑판 모양으로 솎아낸 현 프레임의 픽셀들과 참조 프레임의 픽셀들이 연속 데이터로서 제2 메모리에 저장되기 때문에, 참조 블록이 각 위치에 대하여 제2 메모리로 이동된 때, 움직임 벡터가 2 픽셀 스텝에서 검색된다. 움직임 벡터가 2 픽셀 스텝에서 대략 검색 처리에 의해 구해진 후, 제1 메모리를 이용하여, 1 픽셀 스텝에서 미세 검색 처리가 대략 검색 처리에서 구해진 지점 부근에서 수행된다.

따라서, 바둑판 모양으로 구해진 현 프레임의 픽셀들과 참조 프레임의 픽셀들이 연속 데이터로서 메모리에 저장된 때, 움직임 벡터가 2 픽셀 스텝에서 검색되므로, 대수 검색 처리가 용이하게 수행될 수 있고, 또한 MMX 명령을 이용할 수 있다.

도 5에 나타낸 단계 S2와 같은 움직임 벡터 산출 처리는 블록 매칭 처리에 의해 수행된다. 블록 매칭 처리에서, 움직임 벡터가 블록 매칭 처리에 의해 구해진다. 한편, 처리될 현 프레임으로부터 분리되는 블록으로서 동일한 크기 및 동일한 원정을 갖는 블록이 참조 프레임으로부터 추출된다. 참조 프레임의 블록이 선정된 검색 영역으로 이동되면서, 참조 프레임의 블록의 픽셀들과 현 프레임의 관련 블록의 픽셀들 간의 차분값의 절대값의 합이 잔여분으로서 구해진다. 최소 잔여분을 갖는 참조 프레임의 블록이 구해진다. 따라서, 통상 현 프레임의 블록의 픽셀들과 참조 프레임의 관련 블록의 픽셀들 간의 차분값의 절대값의 합으로서 잔여분이 구해진다. 그러나, 산출 단계의 수가 커지기 때문에, 움직임 벡터 산출 처리가 고속으로 수행될 수 없다.

따라서, 본 발명의 제4 실시예에 따르면, 현 프레임의 블록의 윤곽 픽셀(contour pixel)과 참조 프레임의 관련 블록의 윤곽 픽셀 간의 차분값의 절대값의 합을 산출함으로써 잔여분이 구해진다.

한편, 도 25에서, 참조 프레임의 하나의 블록은 (16×16) 픽셀들로 구성된다. 또한, 현 프레임의 하나의 블록은 (16×16) 픽셀들로 구성된다. 참조 프레임의 각 픽셀의 값은 P(Hr, Vr)로 표기된다. 또한, 현 프레임의 각 픽셀의 값은 P(Hc, Vc)로 표기된다. 참조 프레임의 블록의 상측 윤곽 픽셀들 P(Hr, Vr) 내지 P(Hr+15, Vr)과, 현 프레임의 관련 블록의 상측 윤곽 픽셀들 P(Hc, Vc) 내지 P(Hc+15, Vc) 간의 차분값의 절대값의 합이 잔여분으로서 구해진다. 참조 프레임의 블록의 좌측 윤곽 픽셀들 P(Hr, Vr+1) 내지 P(Hr, Vr+14)와 현 프레임의 관련 블록의 좌측 윤곽 픽셀들 P(Hc, Vc+1) 내지 P(Hc, Vc+14) 간의 차분값의 절대값의 합이 잔여분으로서 구해진다. 참조 프레임의 블록의 우측 윤곽 픽셀들 P(Hr+15, Vr+1) 내지 P(Hr+15, Vr+14)와 현 프레임의 관련 블록의 우측 윤곽 픽셀들 P(Hc+15, Vc+1) 내지 P(Hc+15, Vc+14) 간의 차분값의 절대값의 합이 잔여분으로서 구해진다. 참조 프레임의 블록의 하측 윤곽 픽셀들 P(Hr, Vr+15) 내지 P(Hr+15, Vr+15)와, 현 프레임의 관련 블록의 하측 윤곽 픽셀들 P(Hc, Vc+15) 내지 P(Hc+15, Vc+15) 간의 차분값의 절대값의 합이 잔여분으로서 구해진다.

도 26은 잔여분을 구하기 위하여, 현 프레임의 블록의 윤곽 픽셀들과 참조 프레임의 관련 블록의 윤곽 픽셀들 간의 차분값의 절대값의 합을 구하는 처리를 나타낸 순서도이다.

도 26을 참조하면, 축적값 AD의 값이 '0'으로 초기화된다(단계 S221). 그 후, 현 프레임의 픽셀의 수평 위치 Hc 및 수직 위치 Vc와 참조 프레임의 픽셀의 수평 위치 Hr 및 수직 위치 Vr이 초기화된다(단계 S222). 오프셋 O가 '0'으로 초기화된다(단계 S223).

참조 프레임의 픽셀 P(Hr+O, Vr)의 값과 현 프레임의 픽셀 P(Hc+O, Vc)의 값 간의 차분값의 절대값이 축적값 AD로서 구해진다(단계 S224). 그 후, 오프셋 O가 증분된다(단계 S225). 그 후, 오프셋 O가 16보다 작은지의 여부가 판정된다(단계 S226). 단계 S226에서 판정된 결과가 '예'인 경우(즉, 오프셋 O가 16보다 작은 경우), 단계 S224로 복귀한다.

단계 S224에서 단계 S226까지의 루프에서, 참조 프레임의 블록의 상측 윤곽 픽셀들 P(Hr, Vr) 내지 P(Hr+15, Vr)와, 현 프레임의 관련 블록의 상측 윤곽 픽셀들 P(Hc, Vc) 내지 P(Hc+15, Vc) 간의 차분값의 절대값의 합이 구해진다.

즉, 오프셋 O가 단계 S223에서 '0'으로 초기화되기 때문에, 참조 프레임의 블록의 좌상단 픽셀들 P(Hr, Vr)의 값과 현 프레임의 관련 블록의 좌상단 픽셀 P(Hc, Vc)의 값 간의 차분값의 절대값이 축적값 AD로서 구해진다.

그 후, 단계 S225에서 오프셋 O가 증분된다. 따라서, 참조 프레임의 블록의 픽셀 P(Hr+1, Vr)의 값과 현 프레임의 관련 블록의 픽셀 P(Hc+1, Vc)의 값 간의 차분값의 절대값이 구해지고 축적값 AD에 가산된다. 상기 루프의 단계들은 오프셋 O가 '15'가 될 때까지 반복된다. 따라서, 참조 프레임의 블록의 상측 윤곽 픽셀들 P(Hr, Vr) 내지 P(Hr+15, Vr)와 현 프레임의 관련 블록의 상측 윤곽 픽셀들 P(Hc, Vc) 내지 P(Hc+15, Vc) 간의 차분값의 절대값의 합이 구해진다.

따라서, 단계 S224에서 단계 S226까지의 루프에서, 현 프레임의 블록의 상측 윤곽 픽셀들과 참조 프레임의 관련 블록의 상측 윤곽 픽셀들 간의 차분값의 절대값의 합이 구해진다. 그 후, 단계 S226에서 오프셋 O가 '16'인지의 여부가 판정된다. 단계 S226에서 판정된 결과가 '아니오'인 경우(즉, 오프셋 O가 '16'인 경우), 블록의 우단이 검출된 것이므로, 오프셋 O는 '1'로 초기화된다(단계 S227).

그 후, 참조 프레임의 블록의 픽셀 P(Hr, Vr+O)의 값과 현 프레임의 관련 블록의 픽셀 P(Hc, Vc+O)의 값 간의 차분값이 구해진다. 참조 프레임의 블록의 픽셀 P(Hr+15, Vr+O)의 값과 현 프레임의 관련 블록의 픽셀 P(Hc+15, Vc+O)의 값 간의 차분값이 구해진다. 이들 절대값의 합이 축적값 AD로서 구해진다(단계 S228). 그 후, 오프셋 O가 증분된다(단계 S229). 그 후, 오프셋 ORK '15'보다 작은지의 여부가 판정된다(단계 S230). 단계 S230에서 판정된 결과가 '예'인 경우(즉, 오프셋 O가 '15'보다 작은 경우), 단계 S228로 복귀한다.

단계 S228에서 단계 S230까지의 루프에서, 참조 프레임의 블록의 좌측 윤곽 픽셀들 P(Hr, Vr+1) 내지 P(Hr, Vr+14)와 현 프레임의 관련 블록의 좌측 윤곽 픽셀들 P(Hc, Vc+1) 내지 P(Hc, Vc+14) 간의 차분값의 절대값의 합이 구해진다. 또한, 참조 프레임의 블록의 우측 윤곽 픽셀들 P(Hr+15, Vr+1) 내지 P(Hr+15, Vr+14)와 현 프레임의 관련 블록의 우측 윤곽 픽셀들 P(Hc+15, Vc+1) 내지 P(Hc+15, Vc+14) 간의 차분값의 절대값의 합이 구해진다.

그 후, 단계 S230에서 오프셋 O가 '15'인지의 여부가 판정된다. 단계 S230에서 판정된 결과가 '아니오'인 경우(즉, 오프셋 O가 '15'인 경우), 블록의 하단이 검출된 것이므로, 오프셋 O는 '0'으로 초기화된다(단계 S231).

다음에, 참조 프레임의 블록의 픽셀 P(Hr+O, Vr+15)와 현 프레임의 관련 블록의 픽셀 P(Hc+O, Vc+15) 간의 차분값의 절대값이 축적값 AD로서 구해진다(단계 S232). 그 후, 오프셋 O가 증분된다(단계 S233). 그 후, 오프셋 O가 '16'보다 작은지의 여부가 판정된다(단계 S234). 단계 S234에서 판정된 결과가 '예'인 경우(즉, 오프셋 O가 '16'보다 작은 경우), 단계 S232로 복귀한다.

단계 S232에서 단계 S234까지의 루프에서, 참조 프레임의 블록의 하측 윤곽 픽셀들 P(Hr, Vr+15) 내지 P(Hr+15, Vr+15)와 현 프레임의 관련 블록의 하측 윤곽 픽셀들 P(Hc, Vc+15) 내지 P(Hc+15, Vc+15) 간의 차분값의 절대값의 합이 구해진다. 단계 S234에서 판정된 결과가 '아니오'인 경우(즉, 오프셋 O가 '16'인 경우), 블록의 우단이 검출된 것이므로, 처리는 완

요된다.

단계 S224에서 단계 S226까지의 루프에서, 현 프레임의 블록의 상측 윤곽 픽셀들과 참조 프레임의 관련 블록의 상측 윤곽 픽셀들 간의 차분값의 절대값의 합이 구해진다. 단계 S228에서 단계 S230까지의 루프에서, 현 프레임의 블록의 좌측 윤곽 픽셀들과 참조 프레임의 관련 블록의 좌측 윤곽 픽셀들 간의 차분값의 절대값의 합이 구해진다. 또한, 현 프레임의 블록의 우측 윤곽 픽셀들과 참조 프레임의 관련 블록의 우측 윤곽 픽셀들 간의 차분값의 절대값의 합이 구해진다. 단계 S232에서 단계 S234까지의 루프에서, 현 프레임의 블록의 하측 윤곽 픽셀들과 참조 프레임의 관련 블록의 하측 윤곽 픽셀들 간의 차분값의 절대값의 합이 구해진다. 따라서, 현 프레임의 블록의 4변의 윤곽 픽셀들과 참조 프레임의 관련 블록의 4변의 윤곽 픽셀들 간의 차분값의 절대값의 합이 구해진다.

따라서, 현 프레임의 블록의 윤곽 픽셀들과 참조 프레임의 관련 블록의 윤곽 픽셀들 간의 차분값의 절대값의 합이 구해지는 경우, 잔여분을 구하는 산출 단계의 수를 현저하게 줄일 수 있다. 따라서, 블록 매칭 처리가 고속으로 수행될 수 있다. 즉, 블록의 크기가 (16×16) 픽셀인 경우, 하나의 블록의 모든 픽셀들을 산출하기 위해서는 256 감산이 필요하다. 이에 반해, 윤곽 픽셀들을 산출하기 위해서는, 60 감산만이 필요하다. 또한, 윤곽 픽셀들이 속아내지 않으므로, 움직임 벡터는 1 픽셀의 정밀도로 구해질 수 있다.

다음에, 본 발명의 제5 실시예에 따른 화상 엔코딩 장치에 대하여 설명한다.

도 27은 본 발명의 제5 실시예에 따른 화상 엔코딩 장치의 구성을 나타낸다. 도 27을 참조하면, 화상 엔코딩 장치(401)는 프레임 버퍼(202), 움직임 검출부(203), 잔여분 정보 생성부(204), 글로벌 벡터 검출부(205), 및 제어부(206)를 포함한다. 화상 데이터는 프레임 버퍼(202)에 입력된다. 움직임 검출부(203)는 프레임 버퍼(202)에 저장된 화상 데이터의 움직임 성분을 검출한다. 잔여분 정보 생성부(204)는 움직임 잔여분 정보 AD를 생성한다. 글로벌 벡터 검출부(205)는 전체 화상의 움직임 벡터를 검출한다. 제어부(206)는 장치의 각 부분에 대해 엔코딩 처리를 행하기 위한 파라미터 등을 출력한다.

프레임 버퍼(202)는 외부 장치로부터 화상 데이터를 입력하고 프레임 단위로 화상 데이터를 저장한다. 프레임 버퍼(202)는 화상 데이터를 제어부(206)의 제어하에서 선정된 타이밍으로 움직임 검출부(203), 잔여분 정보 생성부(205), 산출부(207)에 출력한다.

글로벌 벡터 검출부(205)는 복수의 매크로 블록으로서 (프레임 버퍼(202)로부터 수신된) 화상 데이터를 샘플링하고 전체 매크로 블록의 움직임 벡터를 검출한다. 즉, 글로벌 벡터 검출부(205)가 전체 매크로 블록의 움직임 벡터를 구하기 때문에, 글로벌 벡터 검출부(205)는 전체 화상의 움직임 벡터(즉, 글로벌 벡터)를 구하여, 검출된 글로벌 벡터를 움직임 검출부(203)에 제공한다.

사실, 도 28에 도시된 바와 같이, 글로벌 벡터 검출부(205)는 한 화상의 다른 지점들에서 복수의 매크로 블록을 추출하여, 추출된 매크로 블록에서 하나의 움직임 벡터를 검출한다. 이 때, 글로벌 벡터 검출부(205)는 추출된 매크로 블록 각각의 움직임 벡터를 평가 함수(evaluation function)에 제공하여 글로벌 벡터를 구하게 된다. 이 글로벌 벡터 검출부(205)는 추출된 매크로 블록의 움직임 벡터의 평균을 산출하는 함수식을 평가 함수로 사용하여 글로벌 벡터를 구한다.

대안적으로, 글로벌 벡터 검출부(205)는 복수의 인접한 매크로 블록을 추출하여 글로벌 벡터를 구할 수 있다. 즉, 글로벌 벡터 검출부(205)는 각 매크로 블록이 (16×16) 픽셀 또는 예를 들어(32×32) 픽셀의 작은 영역 각각에 대해 글로벌 벡터를 구할 수 있다.

또 다른 대안적인 방법으로서, 도 29에 도시된 바와 같이, 글로벌 벡터 검출부(205)는 스크린을 수직으로 분할한 영역 A, B, 및 C에 대한 글로벌 벡터를 얻을 수 있다. 따라서, 영역 A가 원거리 화상으로서 촬영된 산의 화상이고, 영역 C가 근거리 화상으로서 촬영된 꽃의 화상이며, 영역 A 및 영역 C가 패닝(panning)되는 경우, 하나의 스크린이 각각이 움직이는 두 개의 화상을 갖는 경우에도, 각 영역에 대한 글로벌 벡터를 구할 수 있다. 각 영역은 상호 중첩될 수 있다.

움직임 검출부(203)는 프레임 버퍼(202)에 저장된 (16x16 픽셀로 구성된) 화상 데이터 각각의 매크로 블록 움직임 벡터

MV를 검출한다. 움직임 검출부(203)는 참조 프레임의 매크로 블록을 프레임 버퍼(202)로부터 판독된 매크로 블록과 패턴-매칭하여, 움직임 벡터 MV를 검출한다. 움직임 검출부(203)는 검출된 움직임 벡터 MV를 잔여분 정보 생성부(204) 및 제어부(206)에 공급한다. 이때, 움직임 검출부(203)는 글로벌 벡터 검출부(205)로부터 수신된 글로벌 벡터에 의해 움직임 벡터 MV를 생성한다. 즉, 움직임 검출부(203)가 선정된 검색 영역 내에서 각각의 매크럭 블록을 패턴-매칭할 때, 움직임 검출부(203)는 검색 영역 내의 각 매크로 블록을 글로벌 벡터의 오프셋에 의해 변화시키고 움직임 벡터 MV를 얻는다. 움직임 검출부(203)는 글로벌 벡터에 대응하는 검색 영역의 중심 위치를 변화시켜 각 매크로 블록을 패턴-매칭시킨다. 잔여분 정보 생성부(204)는 움직임 검출부(203)로부터 움직임 벡터 MV를 수신한다. 또한, 잔여분 정보 생성부(204)는 프레임 버퍼(202)로부터 화상 데이터의 각 매크로 블록을 수신한다. 움직임 벡터 MV 및 화상 데이터로서, 잔여분 정보 생성부(204)는 움직임 성분들 간의 차분값들의 절대치 합을 잔여분 정보 AD로서 구하고 이 잔여분 정보 AD를 제어부(206)에 공급한다. 제어부(206)는 움직임 검출부(203)로부터 수신된 움직임 벡터 MV와 잔여분 정보 생성부(204)로부터 수신된 움직임 잔여분 정보 AD를 엔코딩 처리하기 위한 매크로 블록 타입을 결정한다. 제어부(206)는 현재 매크로 블록이 예를 들어 화상 타입에 대응하는 인터-매크로 블록인지 또는 인트라-매크로 블록인지를 판정한다. 인터-매크로 블록은 움직임 벡터 MV에 의해 움직임 보상되고 잔여분 정보로 엔코딩되는 매크로 블록이다. 반대로, 인트라 매크로 블록은 움직임 성분없이 단순히 엔코딩되는 매크로 블록이다.

제어부(206)는 스위치(217 및 218)가 결정된 매크로 블록 타입에 대응하여 동작하도록 제어 정보를 생성한다. 또한, 제어부(206)는 움직임 검출부(203)로부터 수신된 움직임 벡터 MV를 움직임 보상부(216)에 제공한다.

또한, 화상 엔코딩 장치(201)는 산출부(207), DCT 처리부(208), 양자화 처리부(209), 가변 길이 코드 엔코딩부(210), 및 버퍼(211)를 포함한다. 산출부(207)는 프레임 버퍼(202)로부터 화상 신호를 수신한다. DCT 처리부(208)는 화상 데이터에 대해 DCT(Discrete Cosine Transform) 처리를 수행한다. 양자화 처리부(209)는 DCT 처리부(208)로부터 수신된 DCT 계수를 양자화시킨다. 가변 길이 코드 엔코딩부(210)는 양자화 처리부(209)로부터 수신된 DCT 계수를 가변 길이 코드로 압축한다. 버퍼(211)는 가변 길이 코드 엔코딩부(210)로부터 수신된 화상 데이터를 저장한다.

DCT 처리부(208)는, 산출부(207)로부터 수신된 화상 데이터의 (8×8) 픽셀의 각 매크로 블록에 대해 2차원 DCT 처리를 수행한다. DCT 처리부(208)는 DCT 계수를 양자화 처리부(209)에 공급한다.

양자화 처리부(209)는 DCT 처리부(208)로부터 수신된 DCT 계수를 각 매크로 블록에 대응하여 변화하는 양자화 스케일로 양자화시킨다. 양자화 처리부(209)는 양자화된 DCT 계수를 가변 길이 코드 엔코딩부(210) 및 역 양자화 처리부(212)로 공급한다.

가변 길이 코드 엔코딩부(210)는 양자화 처리부(209)로부터의 DCT 계수와 제어부(206)로부터의 움직임 벡터 MV를 수신한다. 이러한 정보를 이용하여, 가변 길이 코드 엔코딩부(210)는 엔코딩 처리를 수행한다. 가변 길이 코드 엔코딩부(210)는 MPEG 신택스에 대응하는 가변 길이 코드로 엔코딩 처리를 수행하고, 헤더 처리, 코드 생성 처리 등을 수행하여 화상 데이터를 생성한다. 가변 길이 코드 엔코딩부(210)는 엔코드된 화상 데이터를 버퍼(211)에 공급한다.

버퍼(211)는 가변 길이 코드 엔코딩부(210)로부터 수신된 화상 데이터를 저장하고 이 화상 데이터를 제어부(206)의 제어하에서 선정된 타이밍에서 비트 스트림으로서 출력한다.

또한, 화상 엔코딩 장치(201)는 역 양자화 처리부(212), 역 DCT 처리부(213), 산출부(214), 버퍼(215), 및 움직임 보상부(216)를 포함한다. 역 양자화 처리부(212)는 양자화 처리부(209)로부터 수신된 DCT 계수를 역으로 양자화시킨다. 역 DCT 처리부(213)는 역 양자화 처리부(212)로부터 수신된 DCT 계수에 대한 DCT 처리를 역으로 수행한다. 산출부(214)은 역 DCT 처리부(213)로부터 화상 데이터를 수신한다. 버퍼(215)는 화상 데이터를 저장한다. 움직임 보상부(216)는 버퍼(215)로부터 수신된 화상 데이터를 움직임 보상한다.

역 양자화 처리부(212)는 양자화 처리부(209)로부터 수신된 DCT 계수를 역으로 양자화시킨다. 역 양자화 처리부(212)는 양자화 처리부(209)로부터 수신된 데이터를 양자화 스케일에 따라 역 양자화하고, 그에 의한 DCT 계수를 역 DCT 처리부(213)에 공급한다.

역 DCT 처리부(213)는 역 양자화 처리부(212)로부터 수신된 DCT 계수에 대해 DCT 처리를 수행하여, 그에 의한 DCT 계수를 산출부(214)에 공급한다. 산출부(214)는 역 DCT 처리부(213)에서 처리된 화상 데이터를 수신한다. 또한, 산출부(214)는 (움직임 보상된) 화상 데이터를 스위치(217)를 통해 수신한다. 산출부(214)는 움직임 보상된 화상 데이터와 역 DCT 처리부(213)로부터 수신된 화상 데이터를 합하여, 그에 의한 데이터를 버퍼(215)에 제공한다.

버퍼(215)는 산출부(214)로부터의 화상 데이터의 각 매크로 블록을 수신하고 그 화상 데이터를 저장한다. 움직임 보상부(216)에서 화상 데이터를 움직임 보상할 때, 예측 화상 데이터가 버퍼(215)로부터 판독된다.

움직임 보상부(216)는, 움직임 벡터 MV에 대응하여, 버퍼(215)로부터 예측 화상 데이터의 각 매크로 블록을 판독한다. 화상 엔코딩 장치(201)가 I(인트라) 화상을 생성하는 경우, 각 프레임 버퍼(202) 내에 저장된 화상 데이터의 각 매크로 블록이 산출부(207)를 통해 DCT 처리부(208) 및 양자화 처리부(209)로 공급된다. DCT 처리부(208)는 화상 데이터의 각 매크로 블록에 대해 DCT 처리를 수행한다. 양자화 처리부(209)는 DCT 처리부(208)로부터 수신된 화상 데이터를 양자화시킨다. 가변 길이 코드 엔코딩부(210)는 양자화 처리부(209)로부터 수신된 화상 데이터를 가변 길이 코드에 의해 엔코딩하고 그에 의한 데이터를 버퍼(211)를 통해 비트 스트림으로서 출력한다. 그에 의한 신호는 양자화 처리부(209)에 의해 처리되고, 가변 길이 코드 엔코딩부(210)는 역 양자화 처리부(212) 및 역 DCT 처리부(213)에 의해 화상 데이터로 복원되어, 버퍼(215)에 일시적으로 저장된다.

화상 엔코딩 장치(201)가 P(예측) 화상을 생성하는 경우, 움직임 검출부(203)는 프레임 버퍼(202) 내에 저장된 화상 데이터의 움직임 성분을 검출하여 움직임 벡터 MV를 생성한다. 또한, 잔여분 정보 생성부(204)는 잔여분 정보 AD를 생성한다. 움직임 벡터 MV는 제어부(206)를 통해 움직임 보상부(216)로 공급된다. 움직임 보상부(216)는 버퍼(215) 내에 저장된 화상 데이터에 대해 움직임 보상한다 (I 화상이 생성되는 경우, 그 화상 데이터는 버퍼(215)에 저장됨). 따라서, 움직임 보상부(216)는 각 매크로 블록에 대해 움직임 보상한다. 스위치(217 및 218)는 제어부(206)로부터 수신된 스위치 제어 신호에 따라 폐쇄된다. 산출부(207)는 프레임 버퍼(202)에 움직임 보상부(216)로부터 수신된 예측 화상 데이터를 저장된 화상 데이터로부터 감산한다. DCT 처리부(208)와 양자화 처리부(209)는 상술한 처리를 수행한다. 가변 길이 코드 엔코딩부(210)는 화상 데이터를 엔코딩하여, 그에 의한 데이터를 버퍼(211)를 통해 비트 스트림으로서 출력한다.

화상 엔코딩 장치(201)가 B (양방향 예측) 화상을 생성하는 경우, 움직임 보상부(216)는 버퍼(215) 내에 저장된 선행하는 프레임의 화상 데이터와 다음 프레임의 화상 데이터에 대해 움직임 보상하여 예측 화상 데이터를 생성한다. 산출부(207)는 프레임 버퍼(202) 내에 저장된 화상 데이터로부터 예측 화상 데이터를 감산한다. DCT 처리부(208) 및 양자화 처리부(209)는 상술한 처리를 수행한다. 가변 길이 코드 엔코딩부(210)는 산출부(207)로부터 수신된 데이터를 가변 길이 코드로 엔코딩하고, 그에 의한 데이터를 버퍼(211)를 통해 비트 스트림으로서 출력한다.

도 30은 움직임 벡터 MV를 검출하기 위한 처리 과정을 도시하는 순서도이다. 이러한 처리는 화상 엔코딩 장치(201)에 의해 수행된다.

도 30을 참조하면, 단계 S301에서, 1 프레임의 화상 데이터가 프레임 버퍼(202) 내에 입력된다. 도 30에 도시된 처리에서, 단계 S302 내지 S304에서, 글로벌 벡터가 검출된다. 단계 S305에서, 움직임 검출부(203)는 각 매크로 블록의 움직임 벡터 MV를 생성한다.

단계 S302에서, 글로벌 벡터 검출부(205)는 도 28에 도시된 바와 같이 프레임 버퍼(202) 내에 저장된 각 프레임의 화상 데이터를 입력하고, 화상 데이터로부터 복수의 매크로 블록을 추출한다. 단계 S302에서, 도 29에서 도시된 바와 같이 하나의 스크린을 복수의 영역으로 분할할 수 있고, 복수의 매크로 블록이 이들로부터 추출될 수 있다.

단계 S303에서, 글로벌 벡터 검출부(205)는 단계 S302에서 검출된 각 매크로 블록의 움직임 벡터를 검출한다.

단계 S304에서, 글로벌 벡터 검출부(205)는 각 매크로 블록의 움직임 벡터를 평가 함수에 적용하여 글로벌 벡터를 생성하게 된다. 글로벌 벡터 검출부(205)는 매크로 블록의 움직임 벡터의 평균값을 산출하고 글로벌 벡터를 생성하게 된다.

단계 S305에서, 움직임 검출부(203)는 화상 데이터의 각 매크로 블록을 수신하고, 각 매크로 블록을 단계 S304에서 검출

된 글로벌 벡터와 패턴 매칭시키고, 각 매크로 블록의 움직임 벡터를 검출한다. 이점에서, 움직임 검출부(203)는 글로벌 벡터에 대응하는 검색 영역의 중심 위치를 변화시켜 각 매크로 블록을 패턴-매칭한다.

단계 S306에서, 움직임 검출부(203)는 단계 S305에서의 검출 결과에 대응하는 각 매크로 블록의 움직임 벡터 MV를 검출하고, 이 각 매크로 블록의 움직임 벡터 MV를 잔여분 정보 생성부(204) 및 제어부(206)에 공급한다.

화상 엔코딩 장치(201)에서, 움직임 검출부(203)가 각 매크로 블록의 움직임 벡터 MV를 생성하기 전에, 글로벌 벡터 검출부(205)에서 전체 화상을 하나의 움직임 벡터로 나타내는 글로벌 벡터를 검출한다. 따라서, 움직임 검출부(203)는 광 대역에서 각 매크로 블록의 움직임 벡터 MV를 검출할 필요가 없다. 결과적으로, 움직임 벡터 MV를 검출하기 위한 처리는 감소된 산출 단계로 수행될 수 있다. 즉, 화상 엔코딩 장치(201)에서 움직이고 있는 화상을 패닝하는 경우에도, 글로벌 벡터 검출부(205)로 하여금 전체 화상의 글로벌 벡터를 얻게 할 필요는 없고, 광역 검색 영역에서 각 매크로 블록의 움직임 벡터를 검출할 필요도 없다.

또한, 화상 엔코딩 장치(201)를 사용하여, 화상이 전체 스크린 상에 고속으로 움직이는 경우에서도, 각 매크로 블록의 움직임 벡터를 쉽게 검출할 수 있다.

또한, 화상 엔코딩 장치(201)를 사용하면 하나의 스크린이 복수의 영역으로 분할될 수 있기 때문에, 글로벌 벡터 검출부(5)는 각 영역에 대한 글로벌 벡터를 산출한다. 따라서, 화상이 스크린 상에서 크게 움직이는 경우에서도, 움직임 벡터 MV가 효율적으로 검출될 수 있다.

다음, 본 발명의 제6 실시예를 기술할 것이다.

도 31은 본 발명의 제6 실시예에 따른 화상 엔코딩 장치의 구조를 도시하는 블록도이다.

도 31은 본 발명의 제6 실시예에 따른 화상 엔코딩 장치의 구조를 도시하고 있다. 도 31을 참조하면, 화상 엔코딩 장치(301)는 프레임 버퍼(302), 움직임 검출부(303), 잔여분 정보 생성부(304), 및 제어부(305)를 포함한다. 화상 데이터는 프레임 버퍼(302) 내로 입력된다. 움직임 검출부(303)는 프레임 버퍼(302) 내에 저장된 화상 데이터의 움직임 성분을 검출한다. 잔여분 정보 생성부(304)는 움직임 잔여분 정보 AD를 생성한다. 제어부(306)는 본 장치의 각 부분에 대한 엔코딩 처리에 필요한 파라미터 등을 출력한다.

프레임 버퍼(302)는 외부 장치로부터의 화상 데이터를 입력하고, 화상 데이터를 프레임 단위로 저장한다. 프레임 버퍼(302)는 제어부(305)의 제어에 따라 선정된 시간 간격으로 화상 데이터를 움직임 검출부(303), 잔여분 정보 생성부(304), 및 산출부(307)로 출력한다.

움직임 검출부(303)는 프레임 버퍼(302) 내에 저장된 화상 데이터의 각 매크로 블록(16 x 16 픽셀)의 움직임 벡터 MV를 검출한다. 움직임 검출부(303)는 프레임 버퍼(302)로부터 판독된 매크로 블록을 갖는 참조 프레임의 매크로 블록을 패턴-매칭하고 움직임 벡터 MV를 검출한다. 움직임 검출부(303)는 검출된 움직임 벡터 MV를 잔여분 정보 생성부(304) 및 제어부(305)에 제공한다.

잔여분 정보 생성부(304)는 움직임 검출부(303)로부터 움직임 벡터 MV를 수신한다. 또한, 잔여분 정보 생성부(304)는 프레임 버퍼로 부터 각 매크로 블록의 화상 데이터를 수신한다. 움직임 벡터 MV 및 화상 데이터에 의해, 잔여분 정보 생성부(304)에서는 움직임 성분들 간의 차분값의 절대값의 합을 잔여분 정보 AD로서 구하고, 이 잔여분 정보 AD를 제어부(305) 및 스킵 제어부(310)로 공급한다.

제어부(305)는 움직임 검출부(303)로부터 수신된 움직임 벡터 MV와, 잔여분 정보 생성부(304)로부터 수신된 움직임 잔여분 정보 AD에 대해 엔코딩 처리를 위한 매크로 블록 타입을 결정한다. 제어부(305)는 현재 매크로 블록이 예를 들어 화상 데이터에 대응하는 인터-매크로 블록인지 인트라-매크로 블록인지 결정한다. 인터-매크로 블록은 움직임 벡터로 움직임 보상되고 잔여분 정보로 엔코딩된 매크로 블록이다. 반대로, 인트라-매크로 블록은 움직임 성분없이 단순히 엔코딩된 매크로 블록이다.

제어부(305)는, 스위치(317 및 318)가 결정된 매크로 블록 타입에 대응하여 동작하게 하는 제어 정보를 생성한다. 또한, 제어부(305)는 움직임 검출부(303)로부터 수신된 움직임 벡터 MV를 움직임 보상부(316)에 공급한다.

또한, 화상 엔코딩 장치(301)는 산출부(306), DCT 처리부(307), 양자화 처리부(308), 가변 길이 코드 엔코딩부(309), 및 버퍼(311)를 포함한다. 산출부(306)에서는 프레임 버퍼(302)로부터 화상 신호를 수신한다. DCT 처리부(307)는 화상 데이터에 대한 DCT 처리를 수행한다. 양자화 처리부(308)는 DCT 처리부(307)로부터 수신된 DCT 계수를 양자화한다. 가변 길이 코드 엔코딩부(309)는 양자화 처리부(308)로부터 수신된 DCT 계수를 가변 길이 코드로 압축한다. 스킵 제어부(310)는 DCT 처리부(307), 양자화 처리부(308), 가변 길이 코드 엔코딩부(309) 등을 제어한다. 버퍼(311)는 엔코드된 화상 데이터를 저장한다.

DCT 처리부(307)는, 산출부(306)로부터 수신된 화상 데이터 (8×8) 픽셀들의 매크로 블록 각각에 대한 2차원 DCT 처리를 수행한다. DCT 처리부(307)는 양자화 처리부(308)에 DCT 계수들을 공급한다.

양자화 처리부(308)는, 대응하는 매크로 블록 각각에 따라 달라지는 양자화 스케일로 DCT 처리부(307)로부터 수신된 DCT 계수를 양자화한다. 양자화 처리부(308)는, 양자화된 DCT 계수를 가변 길이 코드 엔코딩부(309) 및 역 양자화 처리부(312)에 제공한다. 양자화 처리외에, 양자화 처리부(308)는 CBP(Coded Block Pattern)을 생성한다. 양자화 처리부(308)는 CBP를 생성하면, CBP를 나타내는 정보를 가변 길이 코드 엔코딩부(309)에 제공한다.

스킵 제어부(310)는, DCT 처리부(307) 및 양자화 처리부(308)가 DCT 처리 및 양자화 처리를 스킵하도록, 잔여분 정보 생성부(304)로부터 수신된 움직임 잔여분 정보에 대응하여 스킵 제어 신호를 생성한다. 스킵 제어부(310)는 움직임 검출부(303)로부터 움직임 잔여분 정보(AD)를 수신하고, 움직임 잔여분 정보(AD)로 CBP의 값을 예측하여, CBP의 값에 해당하는 DCT 계수를 '0'으로 설정한다. 스킵 제어부(310)가 DCT 계수를 '0'으로 설정한 경우, 스킵 제어부(310)는 움직임 보상부(316), DCT 처리부(307), 양자화 처리부(308), 및 가변 길이 코드 엔코딩부(309)에 스킵 제어 신호를 공급한다. 따라서, 스킵 제어부(310)가 CBP의 값(즉, DCT 계수)을 '0'으로 설정한 경우, 스킵 제어부(310)로 인해 상기 부분들은 그들의 처리를 스킵하게 된다.

스킵 제어부(310)가 DCT 계수를 '0'으로 설정한 경우, 스킵 제어부(310)는 잔여분 정보 생성부(304)로부터 수신된 움직임 잔여분 정보(AD)를 선정된 값과 비교한다. 선정된 값은 예를 들어 사용자에 의해 지정된다. 다시 말해, 움직임 잔여분 정보(AD)가 선정된 값보다 작은 경우, 스킵 제어부(310)는 CBP의 값이 작은 것으로 판단하여, (DCT 계수로 '0'을 대체하는) 스킵 제어 신호를 상기 부분들에 제공한다. 이와 달리, 움직임 잔여분 정보(AD)가 선정된 값 이상인 경우, 스킵 제어부(310)는 스킵 제어 신호를 생성하지 않는다.

대안적으로, 스킵 제어부(310)는 엔코딩 처리에서 얻은 정보로 선정된 값을 결정하여 (그 정보는 가변 길이 코드 엔코딩부(309)의 가변 길이 코드 엔코딩 처리의 비트 속도 및 양자화 처리부(308)의 양자화 처리의 양자화 스케일 등임), 그 선정된 값을 움직임 잔여분 정보(AD)와 비교할 수 있다. 이 때, 스킵 제어부(310)는 매크로 블록 각각에 대한 움직임 잔여분 정보와 선정된 값을 비교하여, 그 비교 결과에 대응하는 스킵 제어 신호를 생성한다.

또 다른 대안적인 방법으로서, 스킵 제어부(310)는 움직임 잔여분 정보(AD) 대신에 각 매크로 블록의 움직임 잔여분 정보(AD)의 평균값(MAD)을 사용할 수 있다. 이러한 경우에, 움직임 보상부(316)에 의해 평균값(MAD)이 생성된다. 스킵 제어부(310)는 움직임 보상부(316)로부터 평균값(MAD)을 수신하고, 평균값(MAD)에 대응하는 DCT 계수를 '0'으로 설정한다. 스킵 제어부(310)의 상세한 동작이 이하 설명된다.

가변 길이 코드 엔코딩부(309)는 양자화 처리부(308)로부터 DCT 계수를 수신하고, 제어부(305)로부터 움직임 벡터(MV)를 수신한다. 상기 정보를 이용하여, 가변 길이 코드 엔코딩부(309)는 엔코딩 처리를 수행한다. 화상 데이터를 생성하기 위해, 가변 길이 코드 엔코딩부(309)는 MPEG 신텍스에 대응하는 가변 길이 코드로 엔코딩 처리를 수행하고, 헤더 처리, 코드 생성 처리 등을 수행한다. 가변 길이 코드 엔코딩부(309)는 엔코딩된 화상 데이터를 버퍼(311)에 공급한다.

가변 길이 코드 엔코딩부(309)는 CBP의 값이 '0'임을 표시하는 정보를 양자화 처리부(308)로부터 수신된다. 움직임 벡터(MV)가 없는 경우, 가변 길이 코드 엔코딩부(309)는 그 매크로 블록 타입에 해당하는 매크로 블록을 스킵할 수 있다.

버퍼(311)는 가변 길이 코드 엔코딩부(309)로부터 수신된 화상 데이터를 저장하고, 제어부(305)의 제어하에서 그 화상 데이터를 선정된 타이밍에서 비트 스트림으로서 출력한다.

게다가, 화상 엔코딩 장치(301)는 또한 역 양자화 처리부(312), 역 DCT 처리부(313), 산출부(314), 버퍼(315), 및 움직임 보상부(316)를 구비한다. 역 양자화 처리부(312)는 양자화 처리부(308)로부터 수신된 DCT 계수를 역으로 양자화한다. 역 DCT 처리부(313)는 역 양자화 처리부(312)로부터 수신된 DCT 계수에 대하여 DCT 처리를 역으로 수행한다. 산출부(314)는 역 DCT 처리부(313)로부터 화상 데이터를 수신한다. 버퍼(315)는 화상 데이터를 저장한다. 움직임 보상부(316)는 버퍼(315)로부터 수신된 화상 데이터를 움직임 보상한다.

역 양자화 처리부(312)는 양자화 처리부(308)로부터 수신된 DCT 계수를 역으로 양자화한다. 역 양자화 처리부(312)는 양자화 처리부(308)로부터 수신된 데이터를 이 처리부의 양자화 스케일로 역 양자화하여, 그 결과로서 생성된 DCT 계수를 역 DCT 처리부(313)에 공급한다.

역 DCT 처리부(313)는 역 양자화 처리부(312)로부터 수신된 DCT 계수에 대하여 DCT 처리를 역으로 수행하여, 그에 따른 DCT 계수를 산출부(314)에 공급한다. 산출부(314)는 역 DCT 처리부(313)에서 처리된 화상 데이터를 수신한다. 또한, 스위치(317)를 통해서 (움직임 보상된) 화상 데이터를 수신한다. 산출부(317)는 움직임 보상된 화상 데이터 및 역 DCT 처리부(313)으로부터 수신된 화상 데이터를 가산하고, 그 결과 데이터를 버퍼(315)에 공급한다.

버퍼(315)는 산출부(314)로부터 화상 데이터를 수신하여, 그 화상 데이터를 저장한다. 움직임 보상부(316)가 화상 데이터를 움직임 보상하는 경우, 예측 화상 데이터가 버퍼(315)로부터 판독된다.

움직임 보상부(316)는 움직임 벡터(MV)에 대응하여 버퍼(315)로부터 예측 화상 데이터의 각 매크로 블록을 판독한다. 움직임 보상부(316)는, 예측 화상 데이터에 대응하여 제어부(305)로부터 수신된 움직임 벡터(MV)를 산출부(306)에 공급한다.

화상 엔코딩 장치(301)가 I (인트라) 화상을 생성한 경우, 프레임 버퍼(302)에 저장된 화상 데이터의 매크로 블록 각각은 산출부(306)을 통해 DCT 처리부(307) 및 양자화 처리부(308)에 공급된다. DCT 처리부(307)는 화상 데이터의 매크로 블록 각각에 대하여 DCT 처리를 수행한다. 양자화 처리부(308)는 DCT 처리부(307)로부터 수신된 화상 데이터를 양자화한다. 가변 길이 코드 엔코딩부(309)는 양자화 처리부(308)로부터 수신된 화상 데이터를 가변 길이 코드로 엔코딩하고, 그 결과 데이터를 버퍼(311)를 통해 비트 스트림으로서 출력한다. 양자화 처리부(308) 및 가변 길이 코드 엔코딩부(309)에 의해 처리된 결과로 생성된 신호는 역 양자화 처리부(312) 및 역 DCT 처리부(313)에 의해 화상 데이터로 복원되어, 버퍼(315)에 일시적으로 저장된다.

화상 엔코딩 장치(301)가 P (예측) 화상을 생성하는 경우, 움직임 검출부(303)는, 움직임 벡터(MV)를 생성하기 위해, 프레임 버퍼(302)에 저장된 화상 데이터의 움직임 성분을 검출한다. 또한, 결과 정보 생성부(304)는 잔여분 정보(AD)를 생성한다. 움직임 벡터(MV)가 제어기(305)를 통해서 움직임 보상부(316)에 공급된다. 움직임 보상부(316)는 버퍼(315)에 저장된 화상 데이터를 움직임 보상한다 (I 화상이 생성된 경우, 화상 데이터는 버퍼(315)에 저장됨). 따라서, 움직임 보상부(316)는 예측 화상을 생성한다. 움직임 보상부(316)는 각 매크로 블록을 움직임 보상한다. 제어부(315)로부터 수신된 스위치 제어 신호에 대응하여 스위치(317 및 318)가 폐쇄된다. 산출부(306)는 프레임 버퍼(302)에 저장된 화상 데이터에서, 움직임 보상부(316)로부터 수신된 예측 화상을 감산한다. DCT 처리부(307) 및 양자화 처리부(308)는 상기 처리들을 수행한다. 가변 길이 코드 엔코딩부(309)는 화상 데이터를 엔코딩하여, 그 결과로 생성된 데이터를 버퍼(311)를 통해 비트 스트림으로서 출력한다.

화상 엔코딩 장치(301)가 B (양방향 예측) 화상을 생성하는 경우, 움직임 보상부(316)는, 예측 화상 데이터를 생성하기 위해, 버퍼(315)에 저장된 이전 프레임의 화상 데이터 및 다음 프레임의 화상 데이터를 움직임 보상한다. 산출부(306)는 프레임 버퍼(302)에 저장된 화상 데이터에서, 예측 화상 데이터를 감산한다. DCT 처리부(307) 및 양자화 처리부(308)가 상기 처리들을 수행한다. 가변 길이 코드 엔코딩부(309)는 가변 길이 코드로 산출부(306)으로부터 수신된 데이터를 엔코딩하고, 그 결과 생성된 데이터를 버퍼(311)를 통해 비트 스트림으로서 출력한다.

도 32는 화상 엔코딩 장치(301)의 엔코딩 처리를 도시하는 순서도이다. 도 32에 도시된 순서도에서, 움직임 벡터(MV)가 프레임 버퍼(302)에 저장된 화상 데이터로부터 검출된다. 검출된 움직임 벡터(MV)에 대응하여, P 화상 또는 B 화상이 생성된다.

단계 S401에서, 한 프레임의 화상 데이터는 프레임 버퍼(302)에 입력된다.

단계 S402에서, 움직임 검출부(303)는 프레임 버퍼(302)에 저장된 화상 데이터의 움직임을 검출하고, 그 움직임을 움직임 벡터(MV)로서 검출한다. 잔여분 정보 생성부(304)는 움직임 벡터(MV)로 움직임 잔여분 정보(AD)를 생성한다. 움직임 벡터(MV) 및 움직임 잔여분 정보(AD)가 제어부(305) 및 스킵 제어부(310)에 각각 공급된다.

단계 S403에서, 스킵 제어부(310)는 결과 정보 생성부(304)로부터 수신된 움직임 잔여분 정보(AD)와 선정된 값을 비교한다. 단계 S403에서 결정된 결과가 '아니오' 인 경우 (즉, 움직임 잔여분 정보(AD)가 선정된 값보다 큼), 흐름은 단계 S404로 진행한다. 이와 달리, 단계 S403에서 결정된 결과가 '예' 인 경우 (즉, 움직임 잔여분 정보(AD)가 선정된 값보다 크지 않음), 흐름은 단계 S407로 진행한다.

단계 S404에서, 제어부(305)로부터 수신된 움직임 벡터(MV)로, 예측 화상 데이터가 생성되고 산출부(306)에 공급된다. 산출부(306)의 산출 결과에 따라서, 움직임 보상부(316)는 화상 데이터의 움직임을 보상한다.

단계 S405에서, 산출부(306)은, 프레임 버퍼(302)로부터 수신된 화상 데이터에서, 단계 S404에서 예측되고 움직임 보상된 화상 데이터를 감산한다. DCT 처리부(307)는 산출부(306)로부터 수신된 화상 데이터에 대해 DCT 처리를 수행한다.

단계 S406에서, 양자화 처리부(308)는, 단계 S405에서 DCT 처리부(307)에 의해 생성된 DCT 계수를 양자화한다.

이와 달리, 단계 S403에서 결정된 결과가 '예' 인 경우 (즉, 움직임 잔여분 정보(AD)가 선정된 값보다 크지 않음), 흐름은 단계 S407로 진행한다. 단계 S407에서, 스킵 제어부(310)는, DCT 처리부(307), 양자화 처리부(308), 가변 길이 코드 엔코딩부(309), 및 움직임 보상부(316)가 그들의 처리를 스킵하게 하는 스킵 제어 신호를 생성하고, 이 스킵 제어 신호를 이들 부분들에 공급한다. 바꾸어 말하면, 버퍼(315)에 저장된 화상 데이터는 가변 길이 코드 엔코딩부(309)에 공급된다.

단계 S408에서, 가변 길이 코드 엔코딩부(309)는 CBP의 값이 '0'인지 아닌지를 판단한다. 단계 S407에서 DCT 계수로 '0'이 설정된 경우, 가변 길이 코드 엔코딩부(309)는 CBP의 값이 '0'이라고 판단한다.

단계 S409에서, 가변 길이 코드 엔코딩부(309)는 예를 들어 헤더 처리 및 가변 길이 코드 생성 처리를 수행하고, 엔코딩된 화상 데이터를 버퍼(311)를 통해 비트 스트림으로서 출력한다. 단계 S408에서 CBP의 값이 '0'으로 결정된 매크로 블록은 (16 × 16) 픽셀의 데이터보다는 헤더, MB 종분, CBP, 벡터 등으로 구성되는 데이터이다. 바꾸어 말하자면, '0'인 CBP 값의 매크로 블록이 결정된 경우, 매크로 블록은 이전 화상과 동일한 화상을 출력을 한다.

화상 엔코딩 장치(301)의 처리에서 (도 32를 참조), 선정된 값과 움직임 잔여분 정보(AD)가 비교되어, DCT 계수의 값이 '0'인지의 여부가 판단된다. 또한, 이 정보는, 선정된 값 대신에 양자화 처리부(308)로부터 수신된 양자화 스케일, 버퍼(315)에 저장된 화상 데이터의 점유량, 비트 속도 등을 이용하여 평가 함수에 따라, DCT 계수가 '0'인지 아닌지를 결정할 수 있다.

화상 엔코딩 장치(301)는, 움직임 잔여분 정보(AD)가 선정된 값과 비교된 비교 결과에 따라, 움직임 보상 처리, DCT 처리, 및 양자화 처리를 스킵하는 스킵 제어부(310)를 가지므로, DCT 계수가 최종적으로 '0'이 되는 화상 데이터의 엔코딩 처리에 필요한 처리 시간이 단축될 수 있다.

또한, 화상 엔코딩 장치(301)에 따르면, 실시간 엔코더 등에서의 DCT 처리 및 양자화 처리에 대한 처리 시간이 탄력적이다. 따라서, 화상 엔코딩 장치(301)가 쉽게 설계될 수 있고, 전력 소비가 감소될 수 있다.

또한, 화상 엔코딩 장치(301)에 따르면, 엔코딩 처리가 소프트웨어에 의해 수행되는 경우에도, 예를 들어 CPU(중앙 처리 장치) 처리의 로드가 감소될 수 있다.

발명의 효과

본 발명에 따르면, 움직임 벡터를 얻은 경우, 참조 프레임의 블록 및 현 프레임의 블록이 주파수 데이터로 직교 변환된다. 화상 데이터가 주파수 데이터로 변환되고, 참조 프레임의 블록 및 현 프레임의 블록 간의 잔여분을 얻은 경우, 산출 단계의 수가 현저히 감소된다. 따라서, 본 처리는 고속으로 수행될 수 있다. 결과적으로, 본 처리는 소프트웨어에 의해 충분히 수행될 수 있다.

본 발명에 따르면, 블록 매칭 산출의 루프에서, 현 프레임 블록의 픽셀들 및 참조 프레임의 관련 블록의 픽셀들 간의 차분값들에 대한 절대값들의 합이 선정된 임계값과 비교된다. 그 합이 임계값을 초과하면, 본 처리는 정지된다. 따라서, 산출 단계의 수가 감소되므로, 움직임 벡터가 고속으로 검색될 수 있다.

임계값의 초기값은 원점에서의 평균 이산 절대값(mean disrete absolute value ; MAD) 및 잔여분 AD (0,0)의 합산된 값으로 설정된다. 임계값이 원점에서의 MAD 및 잔여 AD(0,0)의 합으로 설정되는 경우, 프레임 간 엔코딩 처리가 수행될 때, 움직임 벡터가 확실하게 검출될 수 있다. 반면, 프레임 간 엔코딩 처리가 수행되면, 본 처리는 정지된다. 따라서, 블록 매칭 처리가 효과적으로 수행될 수 있다.

또한, 움직임 벡터가 처음 검색된 경우, 원점에서의 MAD 및 잔여 AD (0,0)의 합에 따라 얻은 임계값이 사용된다. 동일한 블록에 대하여 두번째로 수행되는 블록 매칭 처리에 대한 임계값으로서, 원점에서의 임계값 또는 검출된 합 중 더 작은 값이 사용된다. 바꾸어 말하자면, 이미 얻은 최소값인 임계값이 사용되므로, 움직임 벡터가 효과적으로 검출될 수 있다.

상기한 바와 같이, 본 발명의 움직임 검출 장치 및 움직임 검출 방법에 따르면, 다수의 매크로 블록이 화상으로부터 추출된다. 추출된 매크로 블록들의 움직임 벡터가 검출된다. 검출된 움직임 벡터를 이용하여, 전체 화상의 움직임 벡터가 산출된다. 전체 화상의 움직임 벡터를 이용하여, 매크로 블록 각각의 움직임 벡터가 산출된다. 따라서, 매크로 블록 각각의 움직임 벡터가 산출되기 전에, 전체 화상의 움직임 벡터를 얻을 수 있다. 결과적으로, 본 발명의 움직임 검출 장치 및 움직임 검출 방법에 따르면, 전체 스크린 상에서 크게 이동하는 화상의 움직임 벡터가 쉽게 검출된다. 또한, 화상의 움직임을 검출하기 위한 본 처리에 대한 산출 단계의 수가 현저히 감소될 수 있다.

상기한 바와 같이, 본 발명의 화상 엔코딩 장치 및 화상 엔코딩 방법에 따르면, 화상 데이터의 픽셀 블록에 대한 움직임 벡터가 검출되고, 그로부터 움직임 잔여분 정보가 생성된다. 움직임 잔여분 정보가 선정된 값과 비교된다. 엔코딩 처리에 필요한 선정된 처리가 화상 데이터에 대해 수행된다. 그 결정 결과에 따라, 화상 데이터 처리부의 선정된 처리가 스킵된다. DCT 계수가 최종적으로 '0'이 되는 화상에 대한 엔코딩 처리의 처리 시간이 단축될 수 있다.

본 발명에 따르면, 움직임 벡터가 블록 매칭 처리에 의해 얻어지는 경우, 잔여분은 참조 프레임 블록의 윤곽 픽셀 및 현 프레임의 관련 블록의 윤곽 픽셀 간의 차분값에 대한 절대값의 합으로 얻게 된다. 따라서, 산출 단계의 수가 감소된다. 결과적으로, 본 처리가 고속으로 수행될 수 있다. 참조 프레임 블록의 모든 윤곽 픽셀들 및 현 프레임의 관련 블록의 모든 윤곽 픽셀들 간의 차분값에 대한 절대값의 합이 얻어지므로, 움직임 벡터가 정확하게 검출될 수 있다.

가장 바람직한 형태의 실시예에 관하여 본 발명이 도시되고 설명되었지만, 본 기술 분야에서 숙련자는, 본 발명의 사상 및 범위로부터 벗어나지 않고 본 발명의 형태 및 항목에서 상기의 것 및 여타의 다양한 변경, 생략, 및 부가가 가능하다는 것을 이해하여야 한다.

(57) 청구의 범위

청구항 1. 움직임 벡터 산출 방법에 있어서,

(a) 처리될 현재 화상의 블록에 대응하는 참조 화상으로부터 블록을 추출하는 단계 - 상기 참조 화상의 블록의 크기는 상기 현재 화상의 블록의 크기와 동일하고, 상기 참조 화상의 블록의 원점은 상기 현재 화상의 블록의 원점과 일치함-;

(b) 소정의 검색 영역에서 상기 참조 화상의 블록을 움직이면서, 상기 현재 화상의 블록과 상기 참조 화상의 블록 간의 잔여분을 얻는 단계;

(c) 움직임 벡터를 산출하도록 상기 참조 화상으로부터 최소 잔여분을 갖는 블록을 검출하는 단계;

(d) 상기 참조 화상의 블록의 픽셀 데이터 및 상기 현재 화상의 블록의 픽셀 데이터를 직교 변환하는 단계; 및

(e) 상기 참조 화상의 블록의 직교 변환된 데이터와 상기 현재 화상의 각 블록의 직교 변환된 데이터 간의 잔여분을 얻는 단계

를 포함하는 움직임 벡터 산출 방법.

청구항 2. 제1항에 있어서,

단계(d)는 하다마드 변환 방법에 의해 수행되는 움직임 벡터 산출 방법.

청구항 3. 제1항에 있어서,

상기 참조 화상의 블록의 각각과 상기 현재 화상의 블록을 복수의 블록으로 나누는 단계; 및

상기 나누어진 블록 각각을 직교 변환하는 단계를 더 포함하는 움직임 벡터 산출 방법.

청구항 4. 움직임 벡터 산출 프로그램이 기록된 기록 매체로서, 상기 움직임 벡터 산출 프로그램은 상기 기록 매체를 갖는 시스템으로 하여금 다음의 단계를 수행하게 하는 기록 매체에 있어서, 다음의 단계는

(a) 처리될 현재 화상의 블록에 대응하는 참조 화상으로부터 블록을 추출하는 단계 -상기 참조 화상의 블록의 크기는 상기 현재 화상의 블록의 크기와 동일하고, 상기 참조 화상의 블록의 원점은 상기 현재 화상의 블록의 원점과 일치함-;

(b) 소정의 검색 영역에서 상기 참조 화상의 블록을 움직이면서, 상기 현재 화상의 블록과 상기 참조 화상의 블록 간의 잔여분을 얻는 단계;

(c) 움직임 벡터를 산출하도록 상기 참조 화상으로부터 최소 잔여분을 갖는 블록을 검출하는 단계;

(d) 상기 참조 화상의 블록의 픽셀 데이터 및 상기 현재 화상의 블록의 픽셀 데이터를 직교 변환하는 단계; 및

(e) 상기 참조 화상의 블록의 직교 변환된 데이터와 상기 현재 화상의 각 블럭의 직교 변환된 데이터 간의 잔여분을 얻는 단계

를 포함하는 기록 매체.

청구항 5. 제4항에 있어서, 단계(d)는 하다마드 변환 방법에 의해 수행되는 기록 매체.

청구항 6. 제4항에 있어서,

상기 참조 화상의 블록의 각각과 상기 현재 화상의 블록을 복수의 블록으로 나누는 단계; 및

상기 나누어진 블록 각각을 직교 변환하는 단계를 더 포함하는 기록 매체.

청구항 7. 움직임 벡터 산출 방법에 있어서,

(a) 처리될 현재 화상의 블록에 대응하는 참조 화상으로부터 블록을 추출하는 단계 - 상기 참조 화상의 블록의 크기는 상기 현재 화상의 블록의 크기와 동일하고, 상기 참조 화상의 블록의 원점은 상기 현재 화상의 블록의 원점과 일치함-;

(b) 소정의 검색 영역에서 상기 참조 화상의 블록을 움직이면서, 상기 현재 화상의 블록과 상기 참조 화상의 블록 간의 잔여분을 얻는 단계;

(c) 움직임 벡터를 산출하도록 상기 참조 화상으로부터 최소 잔여분을 갖는 블록을 검출하는 단계;

(d) 상기 참조 화상의 블록의 픽셀과 상기 현재 화상의 블록의 픽셀 간의 잔여분을 산출하면서, 상기 얻어진 잔여분과 소정의 임계값을 비교하는 단계;

(e) 상기 잔여분이 상기 소정의 임계값보다 큰 경우, 상기 움직임 벡터의 산출을 중지하는 단계; 및

(f) 화상의 특성에 대응하는 상기 소정의 임계값의 초기값을 설정하는 단계

를 포함하는 움직임 벡터 산출 방법.

청구항 8. 제7항에 있어서, 상기 소정의 임계값의 상기 초기값은 동일한 화상의 픽셀값과 그 픽셀의 평균값 간의 차분값의 절대값의 합에 대응하여 설정되는 움직임 벡터 산출 방법.

청구항 9. 제7항에 있어서, 상기 소정의 임계값의 상기 초기값은 원점에서의 잔여분에 대응하여 설정되는 움직임 벡터 산출 방법.

청구항 10. 제7항에 있어서, 상기 소정의 임계값의 상기 초기값은 동일한 화상의 픽셀값과 그 픽셀의 평균값 간의 차분값의 절대값의 합 및 원점에서의 잔여분에 대응하여 설정되는 움직임 벡터 산출 방법.

청구항 11. 제7항에 있어서, 상기 소정의 임계값은 현재까지 얻어진 잔여분들중 최소값인 방법.

청구항 12. 움직임 벡터 산출 프로그램이 기록된 기록 매체로서, 상기 움직임 벡터 산출 프로그램은 상기 기록 매체를 갖는 시스템으로 하여금 다음의 단계를 수행하게 하는 기록 매체에 있어서, 다음의 단계는

(a) 처리될 현재 화상의 블록에 대응하는 참조 화상으로부터 블록을 추출하는 단계 - 상기 참조 화상의 블록의 크기는 상기 현재 화상의 블록의 크기와 동일하고, 상기 참조 화상의 블록의 원점은 상기 현재 화상의 블록의 원점과 일치함-;

(b) 소정의 검색 영역에서 상기 참조 화상의 블록을 움직이면서, 상기 현재 화상의 블록과 상기 참조 화상의 블록 간의 잔여분을 얻는 단계;

(c) 움직임 벡터를 산출하도록 상기 참조 화상으로부터 최소 잔여분을 갖는 블록을 검출하는 단계;

(d) 상기 참조 화상의 블록의 픽셀과 상기 현재 화상의 블록의 픽셀 간의 잔여분을 산출하면서, 상기 얻어진 잔여분과 소정의 임계값을 비교하는 단계;

(e) 상기 잔여분이 상기 소정의 임계값보다 큰 경우, 상기 움직임 벡터의 산출을 중지하는 단계; 및

(f) 화상의 특성에 대응하는 상기 소정의 임계값의 초기값을 설정하는 단계

를 포함하는 기록 매체.

청구항 13. 제12항에 있어서, 상기 소정의 임계값의 상기 초기값은 동일한 화상의 픽셀값과 그 픽셀의 평균값 간의

차분값의 절대값의 합에 대응하여 설정되는 기록 매체.

청구항 14. 제12항에 있어서, 상기 소정의 임계값의 상기 초기값은 원점에서의 잔여분에 대응하여 설정되는 기록 매체.

청구항 15. 제12항에 있어서, 상기 소정의 임계값의 상기 초기값은 동일한 화상의 픽셀값과 그 픽셀의 평균값 간의 차분값의 절대값의 합 및 원점에서의 잔여분에 대응하여 설정되는 기록 매체.

청구항 16. 제12항에 있어서, 상기 소정의 임계값은 현재까지 얻어진 잔여분들중 최소값인 기록 매체.

청구항 17. 움직임 검출 장치에 있어서,

화상으로부터 복수의 매크로 블록을 추출하기 위한 추출 수단;

상기 추출 수단에 의해 추출된 상기 복수의 매크로 블록 각각의 움직임 벡터를 검출하기 위한 제1 움직임 검출 수단;

상기 제1 움직임 검출 수단에 의해 검출된 개개의 매크로 블록의 움직임 벡터를 갖는 전체 화상의 움직임 벡터를 산출하기 위한 움직임 산출 수단; 및

상기 움직임 산출 수단에 의해 산출된 움직임 벡터를 갖는 각 매크로 블록의 움직임 벡터를 산출하기 위한 제2 움직임 검출 수단

을 포함하는 움직임 검출 장치.

청구항 18. 제17항에 있어서, 상기 추출 수단은 복수의 인접한 매크로 블록을 추출하는움직임 검출 장치.

청구항 19. 제17항에 있어서, 상기 추출 수단은 전체 화상이 나누어지는 각 영역으로부터 복수의 매크로 블록을 추출하는 움직임 검출 장치.

청구항 20. 움직임 검출 방법에 있어서,

(a) 화상으로부터 복수의 매크로 블록을 추출하는 단계;

(b) 추출된 상기 복수의 매크로 블록 각각의 움직임 벡터를 검출하는 단계;

(c) 검출된 개개의 매크로 블록의 움직임 벡터를 갖는 전체 화상의 움직임 벡터를 산출하는 단계; 및

(d) 산출된 움직임 벡터를 갖는 각 매크로 블록의 움직임 벡터를 산출하는 단계

를 포함하는 움직임 검출 방법.

청구항 21. 제20항에 있어서, 단계(a)는 복수의 매크로 블록을 추출함으로써 수행되는 움직임 검출 방법.

청구항 22. 제20항에 있어서,단계(a)는 전체 화상이 나누어지는 각 영역으로부터 복수의 매크로 블록을 추출함으로써 수행되는 움직임 검출 방법.

청구항 23. 화상 엔코딩 장치에 있어서,

입력 화상 데이터의 소정의 픽셀 블록의 움직임 벡터를 검출하고 움직임 잔여분 정보를 발생시키기 위한 움직임 검출 수단;

상기 움직임 검출 수단으로부터 수신된 움직임 잔여분 정보를 소정값과 비교하고 판정된 결과를 발생시키기 위한 판정 수단;

화상 데이터에 대해 소정의 처리를 수행하기 위한 화상 데이터 처리 수단 - 상기 소정의 처리는 엔코딩 처리를 위해 요구됨-;

화상 데이터에 대해 상기 엔코딩 처리를 수행하기 위한 엔코딩 수단; 및

상기 판정 수단의 판정 결과에 대응하는 상기 화상 데이터 처리 수단에 의해 수행된 소정의 처리를 건너띄고 상기 엔코딩 수단으로 하여금 상기 엔코딩 처리를 수행하게 하는 제어 수단

을 포함하는 화상 엔코딩 장치.

청구항 24. 제23항에 있어서, 상기 움직임 검출 수단은 각 픽셀 블록의 평균 이산 잔여분을 산출하고,

상기 프?수단은 상기 이산 잔여분을 상기 소정값과 비교하는 화상 엔코딩 장치.

청구항 25. 제23항에 있어서, 상기 엔코딩 처리에서 얻어진 정보로 상기 소정값을 판정하기 위한 값 판정 수단을 포함하고,

상기 판정 수단은 상기 값 판정 수단에 의해 판정된 상기 소정값을 갖는 판정된 결과를 발생시키는 화상 엔코딩 장치.

청구항 26. 화상 엔코딩 방법에 있어서,

(a) 입력 화상 데이터의 소정의 픽셀 블록의 움직임 벡터를 검출하고 움직임 잔여분 정보를 발생시키는 단계;

(b) 상기 움직임 잔여분 정보를 소정값과 비교하고 판정된 결과를 발생시키는 단계;

(c) 화상 데이터에 대해 소정의 처리를 수행하는 단계 - 상기 소정의 처리는 엔코딩 처리를 위해 요구됨-;

(d) 상기 판정 결과에 대응하는 상기 소정의 처리를 건너띄고 상기 화상 데이터에 대해 상기 엔코딩 처리를 수행하는 단계

를 포함하는 화상 엔코딩 방법.

청구항 27. 제26항에 있어서,

단계(a)는 각 픽셀 블록의 평균 이산 잔여분을 산출함으로써 수행되고,

단계(b)는 상기 평균 이산 잔여분을 상기 소정값과 비교함으로써 수행되는 화상 엔코딩 방법.

청구항 28. 제26항에 있어서,

상기 엔코딩 처리에서 얻어진 정보로 상기 소정값을 판정하는 단계를 더 포함하고,

단계(b)는 판정된 소정값을 갖는 판정된 결과를 발생시킴으로써 수행되는 화상 엔코딩 방법.

청구항 29. 움직임 벡터 산출 방법에 있어서,

(a) 처리될 현재 화상의 블록에 대응하는 참조 화상으로부터 블록을 추출하는 단계 - 상기 참조 화상의 블록의 크기는 상기 현재 화상의 블록의 크기와 동일하고, 상기 참조 화상의 블록의 원점은 상기 현재 화상의 블록의 원점과 일치함-;

(b) 소정의 검색 영역에서 상기 참조 화상의 블록을 움직이면서, 상기 현재 화상의 블록과 상기 참조 화상의 블록 간의 잔여분을 얻는 단계;

(c) 움직임 벡터를 산출하도록 상기 참조 화상으로부터 최소 잔여분을 갖는 블록을 검출하는 단계;

(d) 상기 현재 화상의 N개의 픽셀과 상기 참조 화상의 N개의 픽셀을 한 번에 추출하는 단계(여기서 N은 정수);

(e) 상기 현재 화상의 N개의 픽셀과 상기 참조 화상의 N개의 픽셀을 연속하는 데이터로서 메모리에 저장하는 단계; 및

(f) 잔여분을 얻도록 상기 메모리로부터 상기 현재 화상의 블록의 픽셀과 상기 참조 화상의 블록의 픽셀을 연속하는 데이터로서 판독하는 단계

를 포함하는 움직임 벡터 산출 방법.

청구항 30. 제29항에 있어서, 상기 잔여분은 복수의 연속하는 데이터 부분이 한 번에 처리되게 하는 명령으로 산출되는 움직임 벡터 산출 방법.

청구항 31. 제29항에 있어서, 상기 현재 화상의 N개의 픽셀과 상기 참조 화상의 N개의 픽셀은 한 번에 바둑판 모양으로(checkerwise) 추출되는 움직임 벡터 산출 방법.

청구항 32. 움직임 벡터 산출 프로그램이 기록된 기록 매체로서, 상기 움직임 벡터 산출 프로그램은 상기 기록 매체를 갖는 시스템으로 하여금 다음의 단계를 수행하게 하는 기록 매체에 있어서, 다음의 단계는

(a) 처리될 현재 화상의 블록에 대응하는 참조 화상으로부터 블록을 추출하는 단계 - 상기 참조 화상의 블록의 크기는 상기 현재 화상의 블록의 크기와 동일하고, 상기 참조 화상의 블록의 원점은 상기 현재 화상의 블록의 원점과 일치함-;

(b) 소정의 검색 영역에서 상기 참조 화상의 블록을 움직이면서, 상기 현재화상의 블록과 상기 참조 화상의 블록 간의 잔여분을 얻는 단계;

(c) 움직임 벡터를 산출하도록 상기 참조 화상으로부터 최소 잔여분을 갖는 블록을 검출하는 단계;

(d) 상기 현재 화상의 N개의 픽셀과 상기 참조 화상의 N개의 픽셀을 한 번에 추출하는 단계(여기서 N은 정수);

(e) 상기 현재 화상의 N개의 픽셀과 상기 참조 화상의 N개의 픽셀을 연속하는 데이터로서 메모리에 저장하는 단계; 및

(f) 잔여분을 얻도록 상기 메모리로부터 상기 현재 화상의 블록의 픽셀과 상기 참조 화상의 블록의 픽셀을 연속하는 데이터로서 판독하는 단계

를 포함하는 기록 매체.

청구항 33. 제32항에 있어서, 상기 잔여분은 복수의 연속하는 데이터 부분이 한 번에 처리되게 하는 명령으로 산출되는 기록 매체.

청구항 34. 제32항에 있어서, 상기 현재 화상의 N개의 픽셀과 상기 참조 화상의 N개의 픽셀은 한 번에 바둑판 모양으

로 추출되는 기록 매체.

청구항 35. 움직임 벡터 산출 방법에 있어서,</P>

(a) 처리될 현재 화상의 블록에 대응하는 참조 화상으로부터 블록을 추출하는 단계 - 상기 참조 화상의 블록의 크기는 상기 현재 화상의 블록의 크기와 동일하고, 상기 참조 화상의 블록의 원점은 상기 현재 화상의 블록의 원점과 일치함-;

(b) 소정의 검색 영역에서 상기 참조 화상의 블록을 움직이면서, 상기 현재 화상의 블록과 상기 참조 화상의 블록 간의 잔여분을 얻는 단계;

(c) 대강의 움직임 벡터를 산출하도록 상기 참조 화상으로부터 최소 잔여분을 갖는 블록을 검출하는 단계;

(d) 단계(c)에서 얻어진 상기 대강의 움직임 벡터 근처에서 상기 참조 화상의 블록을 움직이면서, 상기 현재 화상의 블록과 상기 참조 화상의 블록 간의 잔여분을 얻는 단계;

(e) 미세한 움직임 벡터를 검출하도록 상기 참조 화상으로부터 최소 잔여분을 갖는 블록을 검출하는 단계;

(f) 상기 현재 화상의 픽셀과 상기 참조 화상의 픽셀을 제1 메모리에 저장하는 단계;

(g) 상기 현재 화상의 N개의 픽셀과 상기 참조 화상의 N개의 픽셀을 한 번에 추출하는 단계(여기서 N은 정수); 및

(h) 상기 현재 화상의 N개의 픽셀과 상기 참조 화상의 N개의 픽셀을 연속하는 데이터로서 제2 메모리에 저장하는 단계를 포함하고,

단계(c)는 상기 제2 메모리에 연속하는 데이터로서 저장된 상기 현재 화상의 N개의 픽셀과 상기 참조 화상의 N개의 픽셀로 수행되고,

단계(e)는 상기 제1 메모리에 저장된 상기 현재 화상의 픽셀과 상기 참조 화상의 픽셀로 수행되는 움직임 벡터 산출 방법.

청구항 36. 제35항에 있어서, 상기 잔여분은 복수의 연속하는 데이터 부분이 한 번에 처리되게 하는 명령으로 산출되는 움직임 벡터 산출 방법.

청구항 37. 제36항에 있어서, 상기 현재 화상의 N개의 픽셀과 상기 참조 화상의 N개의 픽셀은 한 번에 바둑판 모양으로 추출되는 벡터 산출 방법.

청구항 38. 움직임 벡터 산출 프로그램이 기록된 기록 매체로서, 상기 움직임 벡터 산출 프로그램은 상기 기록 매체를 갖는 시스템으로 하여금 다음의 단계를 수행하게 하는 기록 매체에 있어서, 다음의 단계는

(a) 처리될 현재 화상의 블록에 대응하는 참조 화상으로부터 블록을 추출하는 단계 - 상기 참조 화상의 블록의 크기는 상기 현재 화상의 블록의 크기와 동일하고, 상기 참조 화상의 블록의 원점은 상기 현재 화상의 블록의 원점과 일치함-;

(b) 소정의 검색 영역에서 상기 참조 화상의 블록을 움직이면서, 상기 현재 화상의 블록과 상기 참조 화상의 블록 간의 잔여분을 얻는 단계;

(c) 대강의 움직임 벡터를 산출하도록 상기 참조 화상으로부터 최소 잔여분을 갖는 블록을 검출하는 단계;

(d) 단계(c)에서 얻어진 상기 대강의 움직임 벡터 근처에서 상기 참조 화상의 블록을 움직이면서, 상기 현재 화상의 블록과 상기 참조 화상의 블록 간의 잔여분을 얻는 단계;

(e) 미세한 움직임 벡터를 검출하도록 상기 참조 화상으로부터 최소 잔여분을 갖는 블록을 검출하는 단계;

(f) 상기 현재 화상의 픽셀과 상기 참조 화상의 픽셀을 제1 메모리에 저장하는 단계;

(g) 상기 현재 화상의 N개의 픽셀과 상기 참조 화상의 N개의 픽셀을 한 번에 추출하는 단계(여기서 N은 정수); 및

(h) 상기 현재 화상의 N개의 픽셀과 상기 참조 화상의 N개의 픽셀을 연속하는 데이터로서 제2 메모리에 저장하는 단계를 포함하고,

단계(c)는 상기 제2 메모리에 연속하는 데이터로서 저장된 현재 화상의 N개의 픽셀과 상기 참조 화상의 N개의 픽셀로 수행되고,

단계(e)는 상기 제1 메모리에 저장된 현재 화상의 픽셀과 상기 참조 화상의 픽셀로 수행되는 기록 매체.

청구항 39. 제38항에 있어서, 상기 잔여분은 복수의 연속하는 데이터 부분이 한 번에 처리되게 하는 명령으로 산출되는 기록 매체.

청구항 40. 제38항에 있어서, 상기 현재 화상의 N개의 픽셀과 상기 참조 화상의 N개의 픽셀은 한 번에 바둑판 모양으로 추출되는 기록 매체.

청구항 41. 움직임 벡터 산출 방법에 있어서,

(a) 처리될 현재 화상의 블록에 대응하는 참조 화상으로부터 블록을 추출하는 단계 - 상기 참조 화상의 블록의 크기는 상기 현재 화상의 블록의 크기와 동일하고, 상기 참조 화상의 블록의 원점은 상기 현재 화상의 블록의 원점과 일치함-;

(b) 소정의 검색 영역에서 상기 참조 화상의 블록을 움직이면서, 상기 현재 화상의 블록과 상기 참조 화상의 블록 간의 잔여분을 얻는 단계;

(c) 움직임 벡터를 산출하도록 상기 참조 화상으로부터 최소 잔여분을 갖는 블록을 塹銖求?단계; 및

(d) 상기 참조 화상의 블록의 윤곽 픽셀과 상기 현재 화상의 블록의 윤곽 픽셀을 비교하여 이들 사이의 잔여분을 얻는 단계

를 포함하는 움직임 벡터 산출 방법.

청구항 42. 제41항에 있어서, 단계(d)는

수평 주사 방향으로 상기 현재 화상의 블록의 상측 윤곽 픽셀과 상기 참조 화상의 블록의 상측 윤곽 픽셀간의 차분값의 절대값을 누산하여 그 합을 구하는 단계;

수직 주사 방향으로 상기 현재 화상의 블록의 좌측 윤곽 픽셀과 상기 참조 화상의 블록의 좌측 윤곽 픽셀간의 차분값의 절대값을 누산하여 그 합을 구하는 단계;

수직 주사 방향으로 상기 현재 화상의 블록의 우측 윤곽 픽셀과 상기 참조 화상의 블록의 우측 윤곽 픽셀간의 차분값의 절대값을 누산하여 그 합을 구하는 단계; 및

수평 주사 방향으로 상기 현재 화상의 블록의 하측 윤곽 픽셀과 상기 참조 화상의 블록의 하측 윤곽 픽셀간의 차분값의 절대값을 누산하여 그 합을 구하는 단계를 포함하는 움직임 벡터 산출 방법.

청구항 43. 움직임 벡터 산출 프로그램이 기록된 기록 매체로서, 상기 움직임 벡터 산출 프로그램은 상기 기록 매체를 갖는 시스템으로 하여금 다음의 단계를 수행하게 하는 기록 매체에 있어서, 다음의 단계는

(a) 처리될 현재 화상의 블록에 대응하는 참조 화상으로부터 블록을 추출하는 단계 - 상기 참조 화상의 블록의 크기는 상기 현재 화상의 블록의 크기와 동일하고, 상기 참조 화상의 블록의 원점은 상기 현재 화상의 블록의 원점과 일치함-;

(b) 소정의 검색 영역에서 상기 참조 화상의 블록을 움직이면서, 상기 현재 화상의 블록과 상기 참조 화상의 블록 간의 잔여분을 얻는 단계;

(c) 움직임 벡터를 산출하도록 상기 참조 화상으로부터 최소 잔여분을 갖는 블록을 검출하는 단계; 및

(d) 상기 참조 화상의 블록의 윤곽 픽셀과 상기 현재 화상의 블록의 윤곽 픽셀을 비교하여 이들 사이의 잔여분을 얻는 단계

를 포함하는 기록 매체.

청구항 44. 제43항에 있어서, 단계(d)는

수평 주사 방향으로 상기 현재 화상의 블록의 상측 윤곽 픽셀과 상기 참조 화상의 블록의 상측 윤곽 픽셀간의 차분값의 절대값을 누산하여 그 합을 구하는 단계;

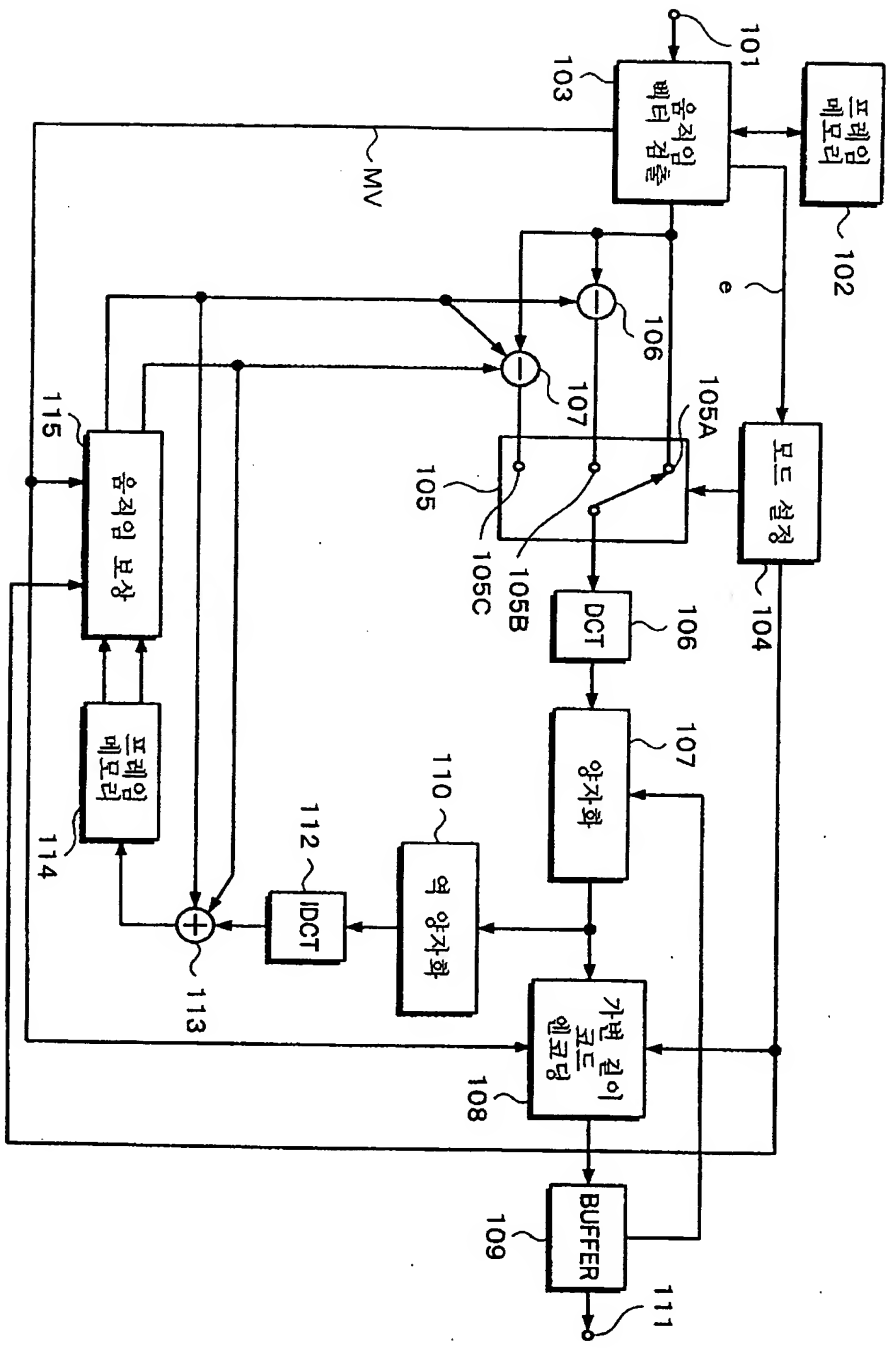
수직 주사 방향으로 상기 현재 화상의 블록의 좌측 윤곽 픽셀과 상기 참조 화상의 블록의 좌측 윤곽 픽셀간의 차분값의 절대값을 누산하여 그 합을 구하는 단계;

수직 주사 방향으로 상기 현재 화상의 블록의 우측 윤곽 픽셀과 상기 참조 화상의 블록의 우측 윤곽 픽셀간의 차분값의 절대값을 누산하여 그 합을 구하는 단계; 및

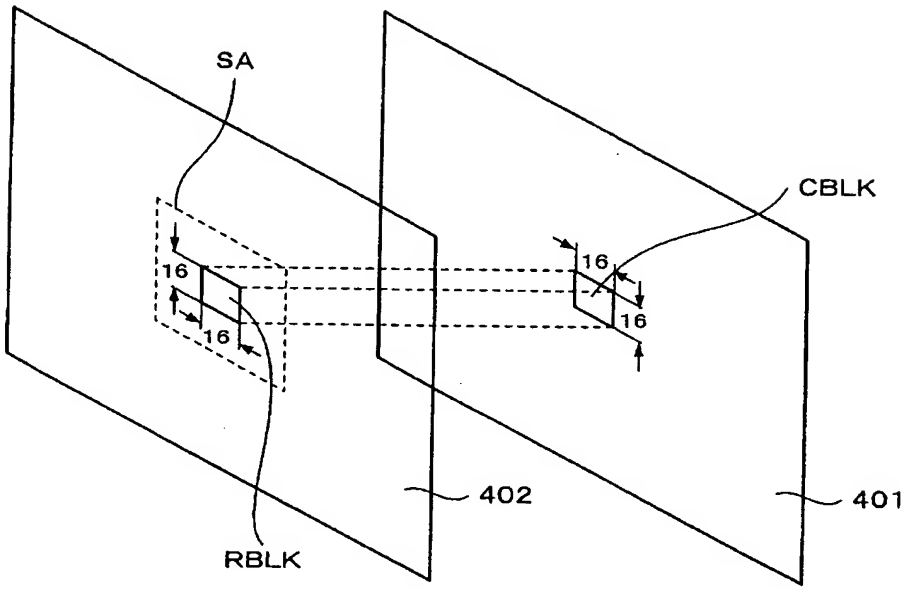
수평 주사 방향으로 상기 현재 화상의 블록의 하측 윤곽 픽셀과 상기 참조 화상의 블록의 하측 윤곽 픽셀간의 차분값의 절대값을 누산하여 그 합을 구하는 단계를 포함하는 기록 매체.

도면

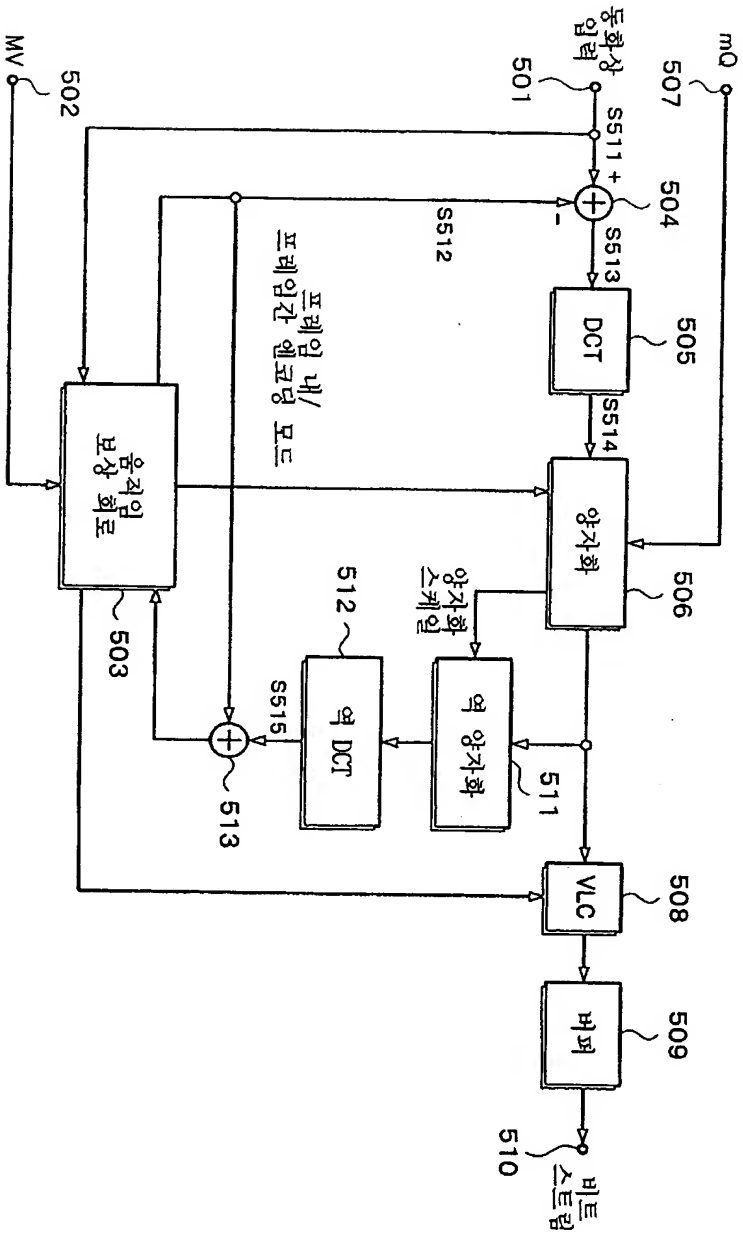
도면1


101
103
움직임 벡터 검출
프레임 메모리
102
MV
e
106
107
105A
115
105
움직임 보상
모드 설정
104
105C
105B
DCT
106
양자화
107
110
112
프레임 메모리
114
역 양자화
IDCT
113
가변 길이 코드 엔코딩
108
BUFFER
109
111

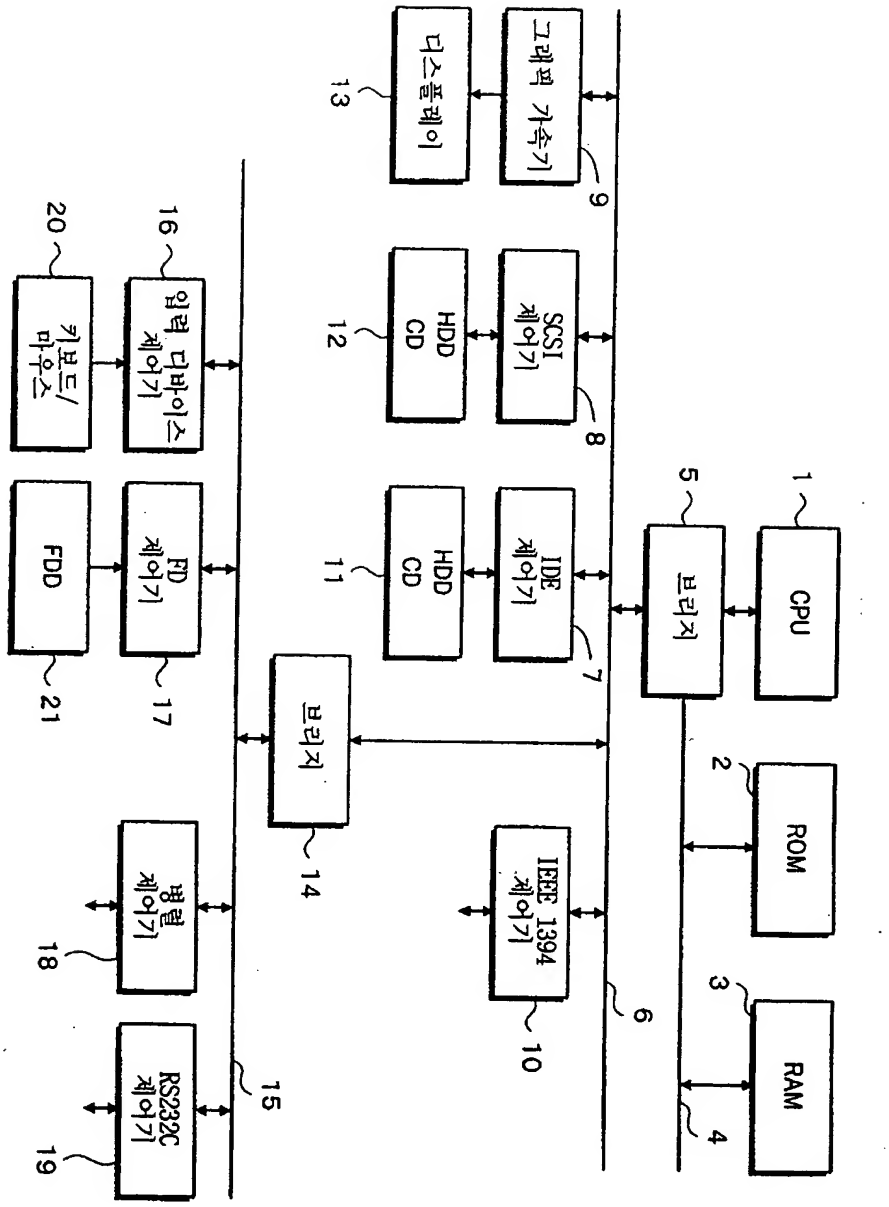
도면2



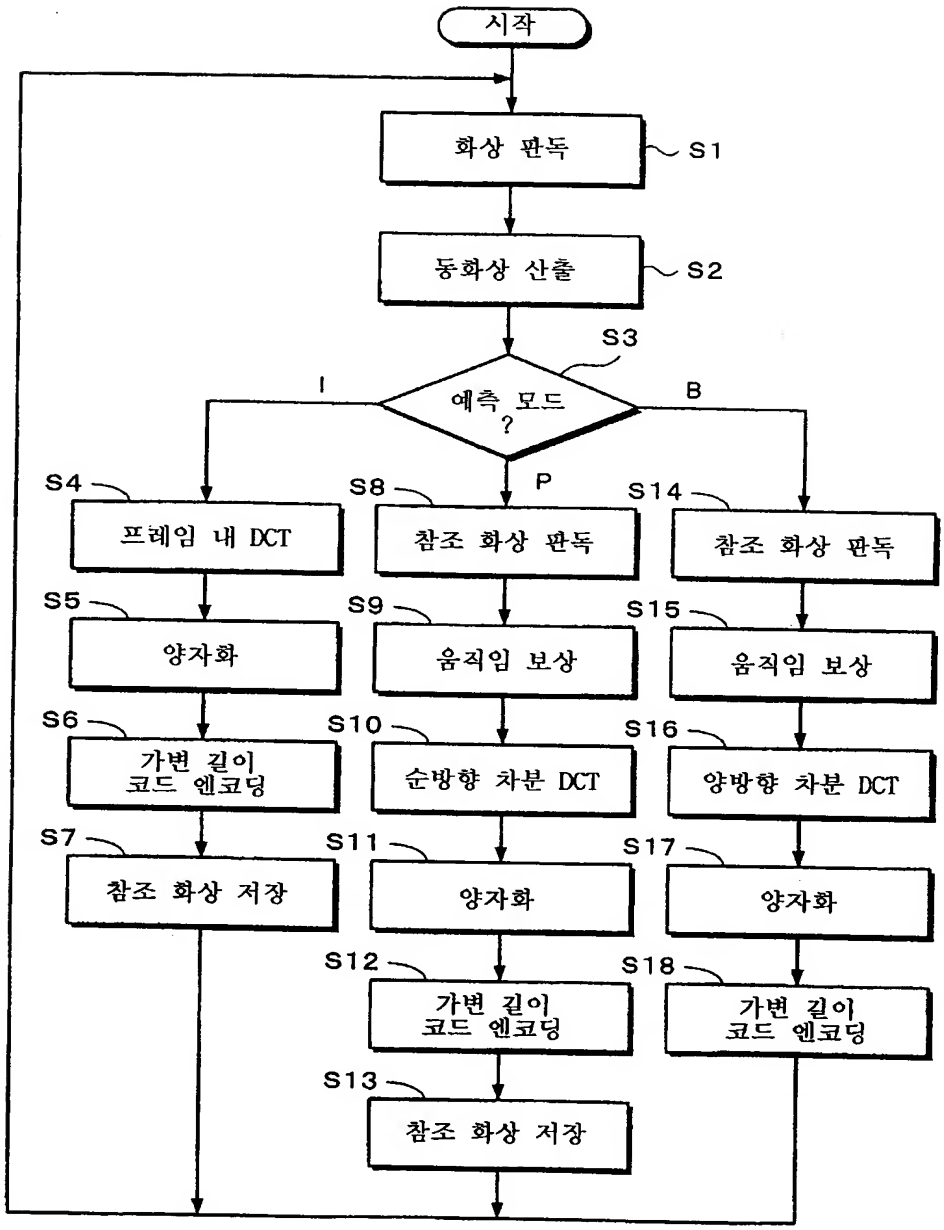
도면3


동화상 입력
501
MV
502
mQ
507
S511
+
-
504
S512
S513
DCT
505
S514
양자화
506
프레임간 엔코딩 모드
움직임 보상 회로
503
양자화 스케일
512
역 DCT
S515
513
역 양자화
511
VLC
508
버퍼
509
510
비트 스트림

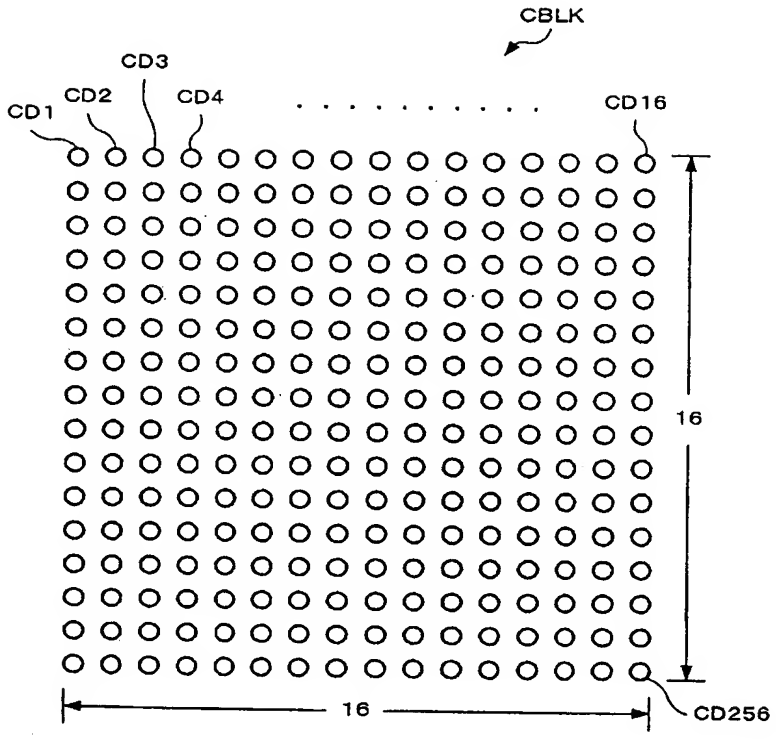
도면4


CPU
1
브리지
5
ROM
2
RAM
3
4
6
IEEE 1394 제어기
10
IDE 제어기
7
HDD CD
11
SCSI 제어기
8
HDD CD
12
그래픽 가속기
9
디스플레이
13
브리지
14
15
RS232C 제어기
19
병렬 제어기
18
FD 제어기
17
FDD
21
입력 디바이스 제어기
16
키보드/ 마우스
20

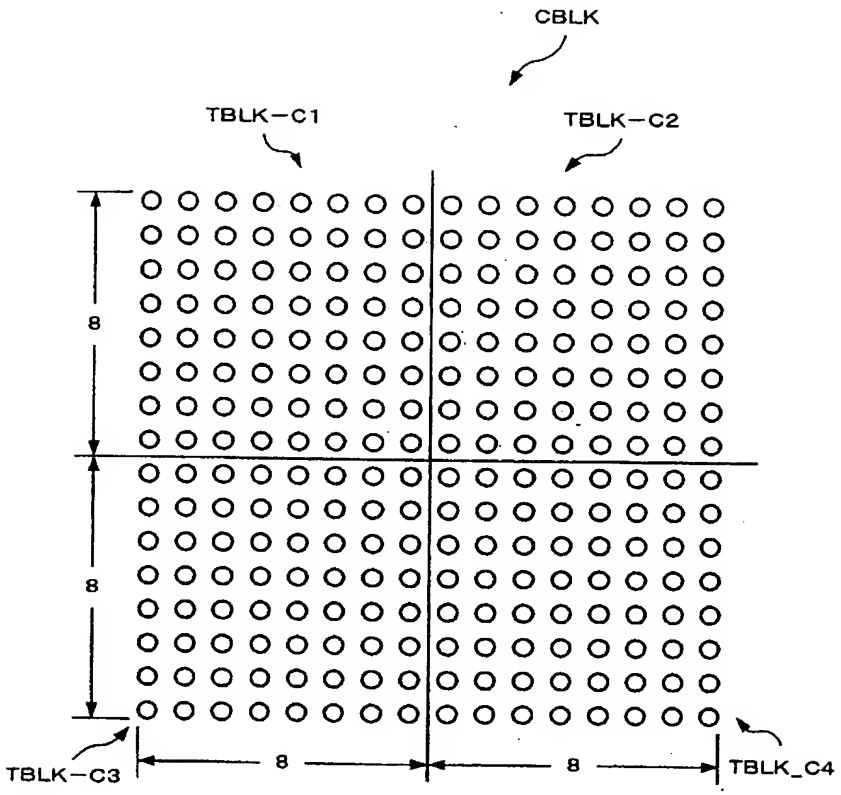
도면5


시작
화상 판독
S1
동화상 산출
S2
S3
예측 모드 ?
I
P
B
S4
프레임 내 DCT
S5
양자화
S6
가변 길이 코드 엔코딩
S7
참조 화상 저장
S8
참조 화상 판독
S9
움직임 보상
S10
순방향 차분 DCT
S11
양자화
S12
가변 길이 코드 엔코딩
S13
참조 화상 저장
S14
참조 화상 판독
S15
움직임 보상
S16
양방향 차분 DCT
S17
양자화
S18
가변 길이 코드 엔코딩

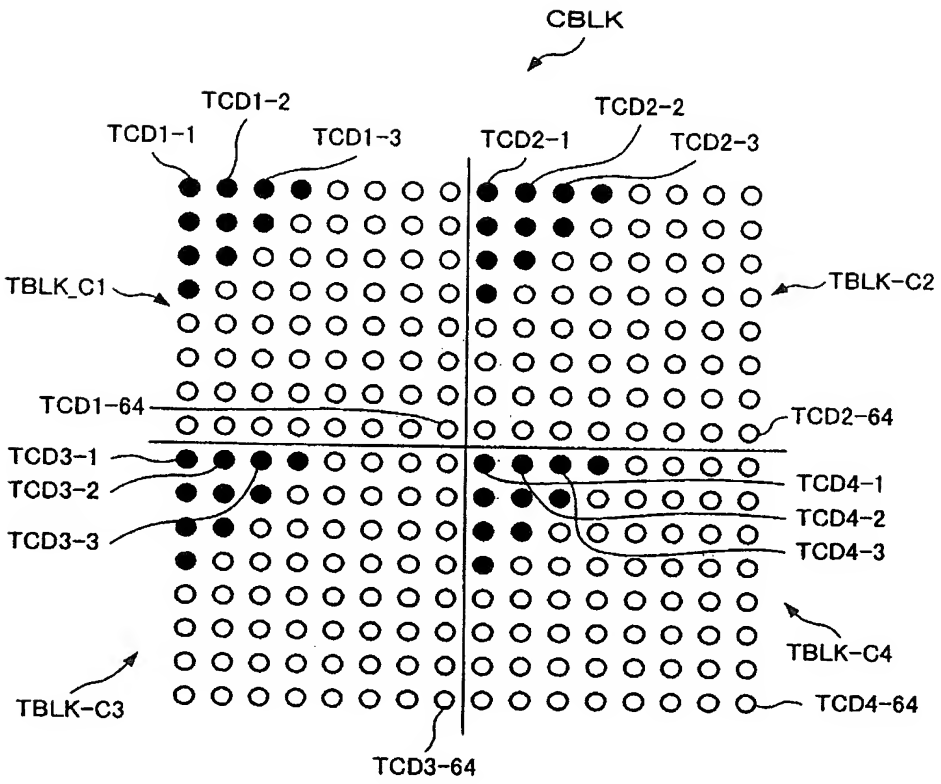
도면6



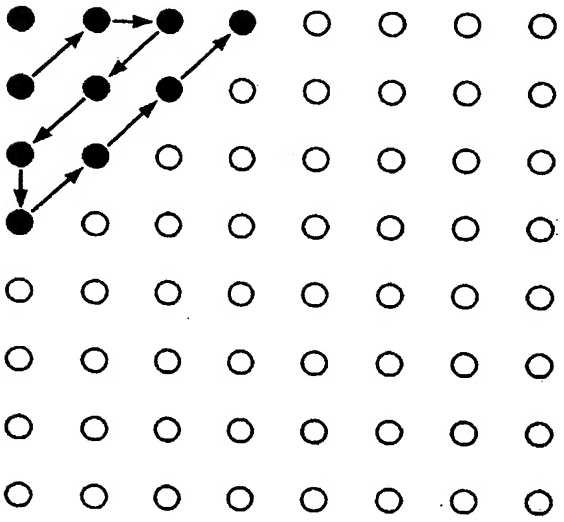
도면7



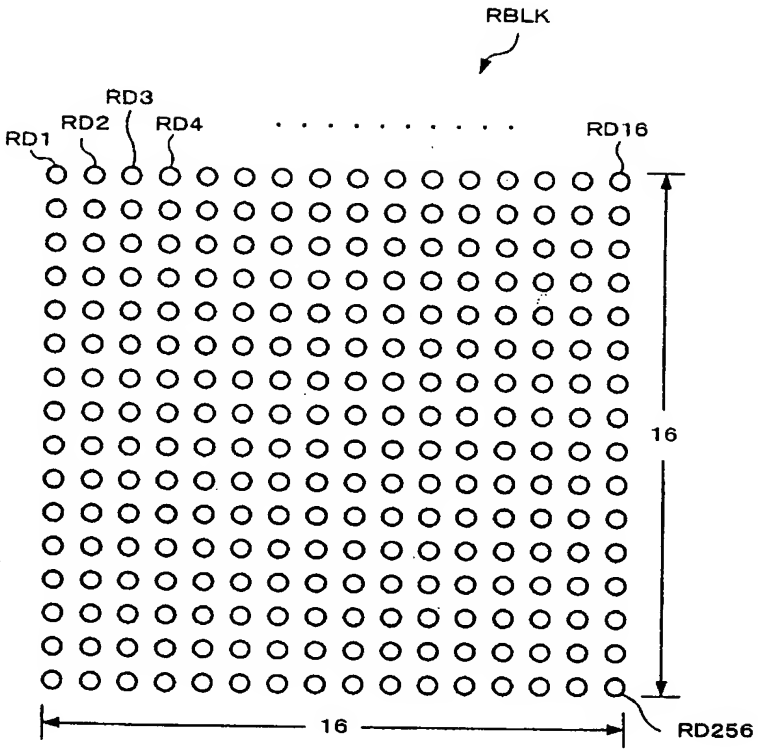
도면8


CBLK
TCD1-2
TCD1-1
TCD1-3
TCD2-1
TCD2-2
TCD2-3
TBLK_C1
TBLK-C2
TCD1-64
TCD2-64
TCD3-1
TCD3-2
TCD3-3
TCD4-1
TCD4-2
TCD4-3
TBLK-C4
TBLK-C3
TCD4-64
TCD3-64

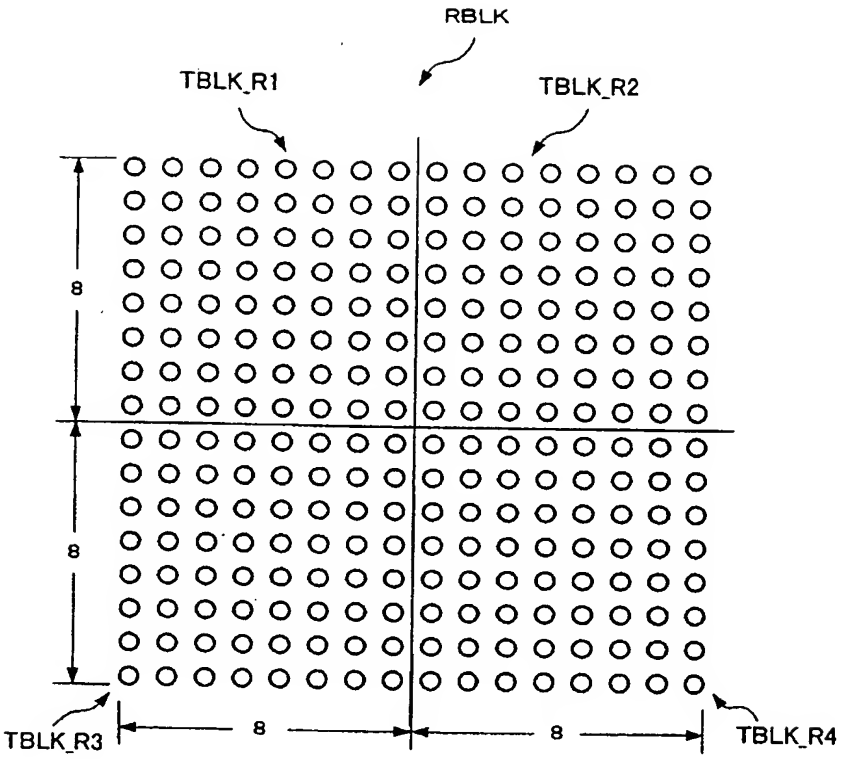
도면9



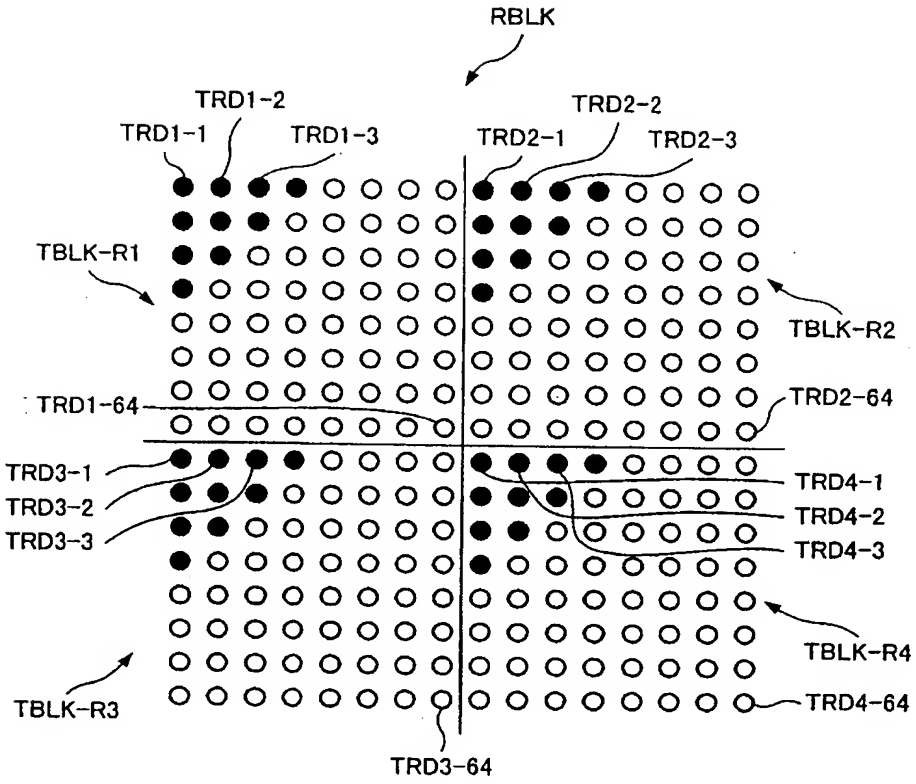
도면10



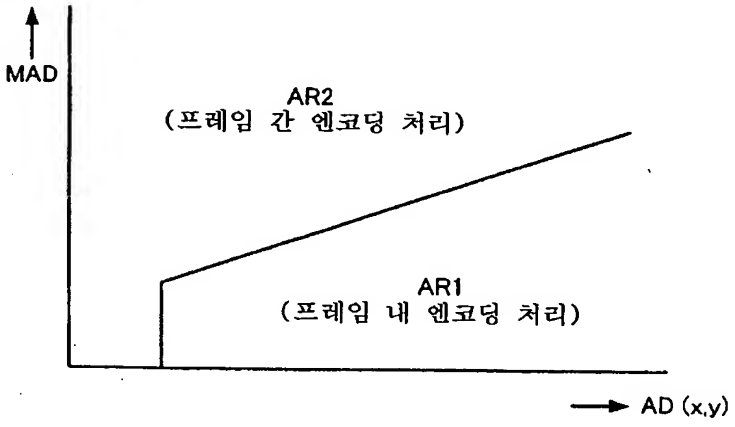
도면11



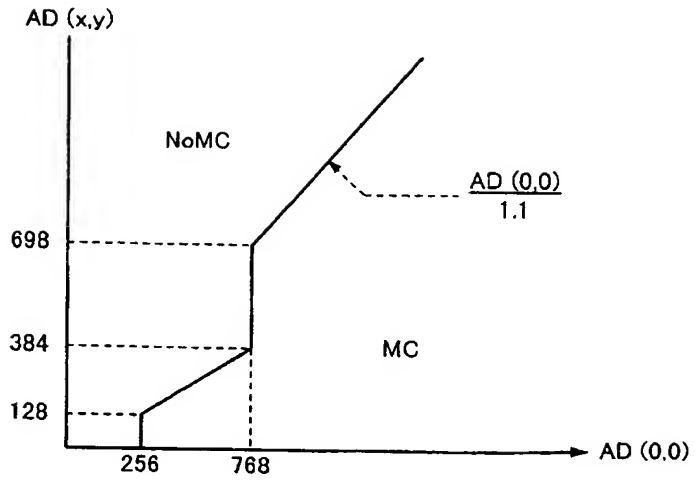
도면12


RBLK
TRD1-2
TRD1-1
TRD1-3
TRD2-2
TRD2-1
TRD2-3
TBLK-R1
TBLK-R2
TRD1-64
TRD2-64
TRD3-1
TRD3-2
TRD3-3
TRD4-1
TRD4-2
TRD4-3
TBLK-R4
TBLK-R3
TRD4-64
TRD3-64

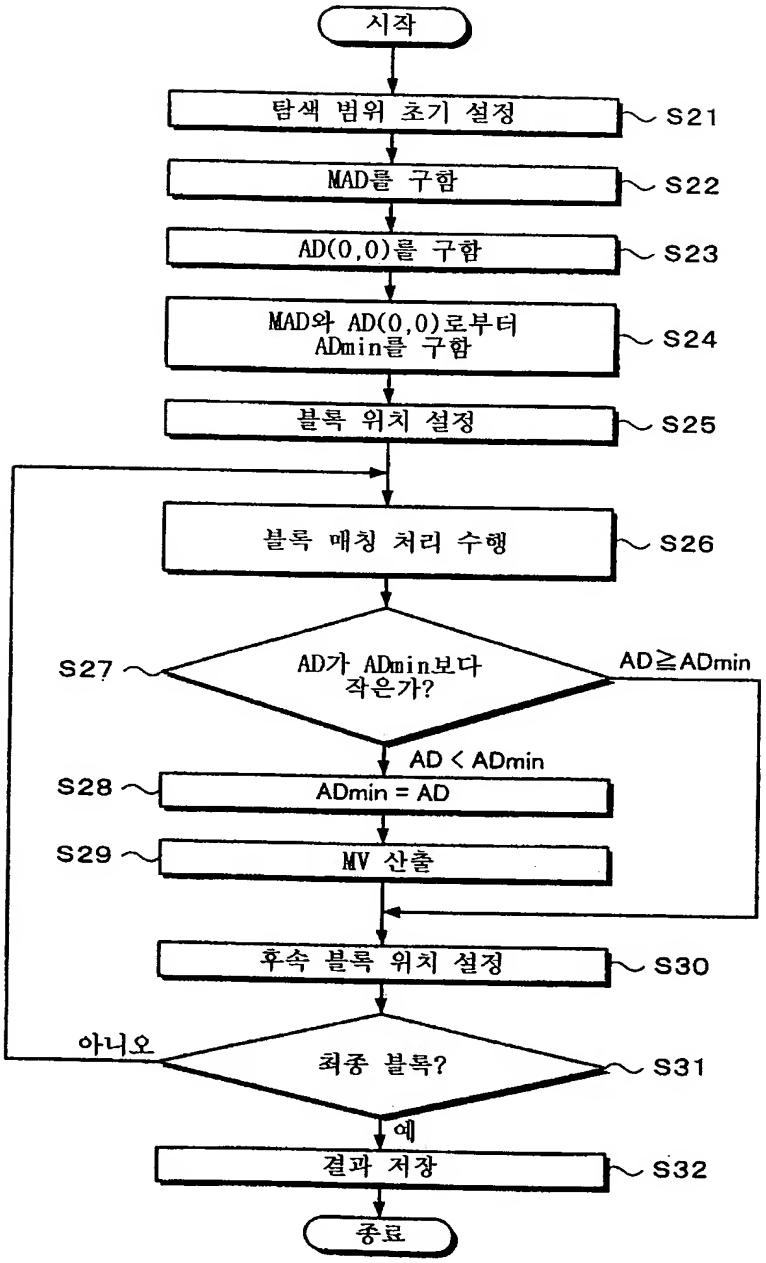
도면13


MAD
AR2
(프레임 간 엔코딩 처리)
AR1
(프레임 내 엔코딩 처리)
AD (x,y)

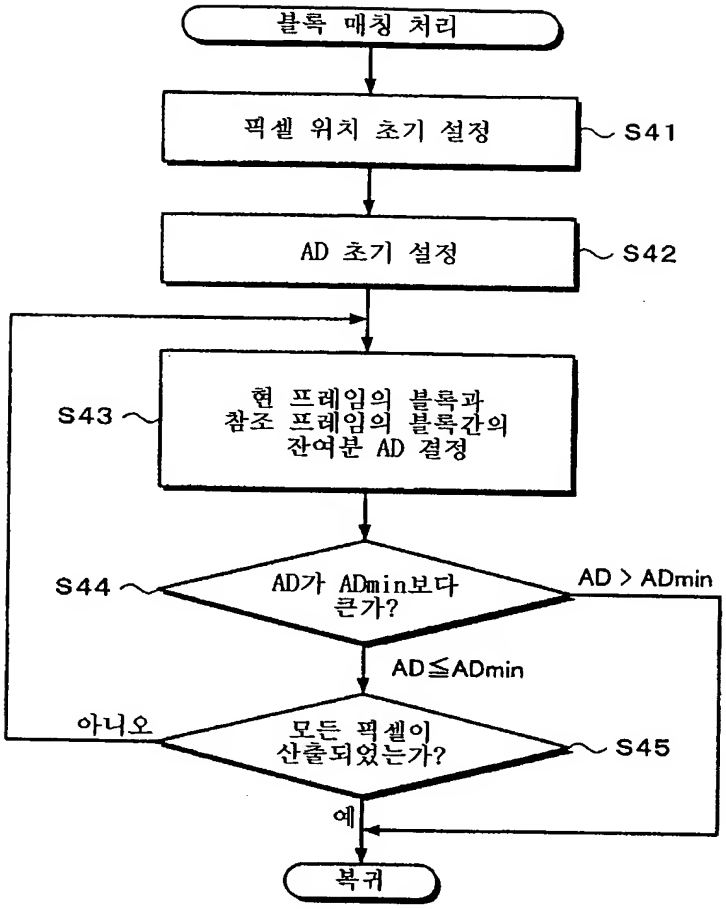
도면14



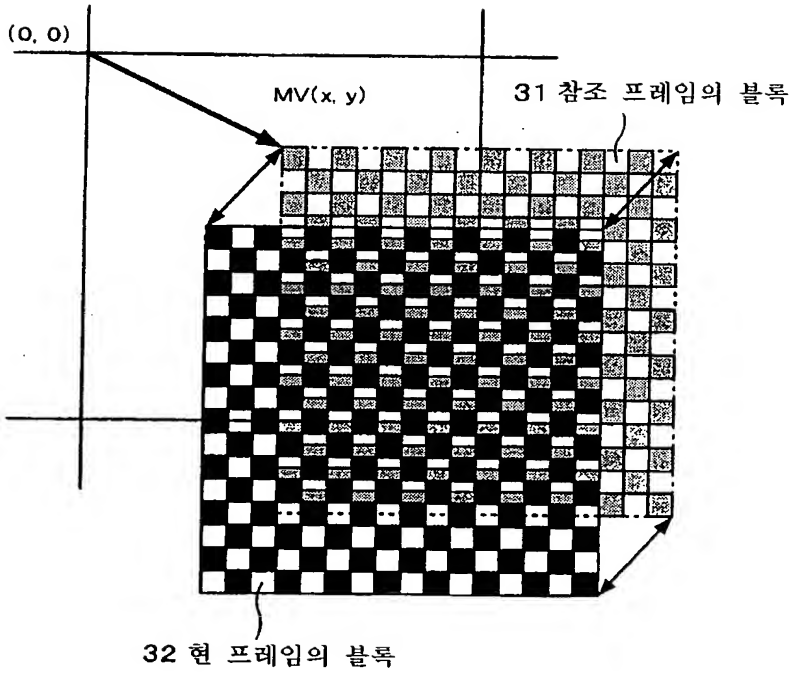
도면15


시작
탐색 범위 초기 설정
S21
MAD를 구함
S22
AD(0,0)를 구함
S23
MAD와 AD(0,0)로부터
ADmin를 구함
S24
블록 위치 설정
S25
블록 매칭 처리 수행
S26
S27
AD가 ADmin보다
작은가?
AD≧ADmin
AD < ADmin
S28
ADmin = AD
S29
MV 산출
후속 블록 위치 설정
S30
아니오
최종 블록?
S31
예
결과 저장
S32
종료

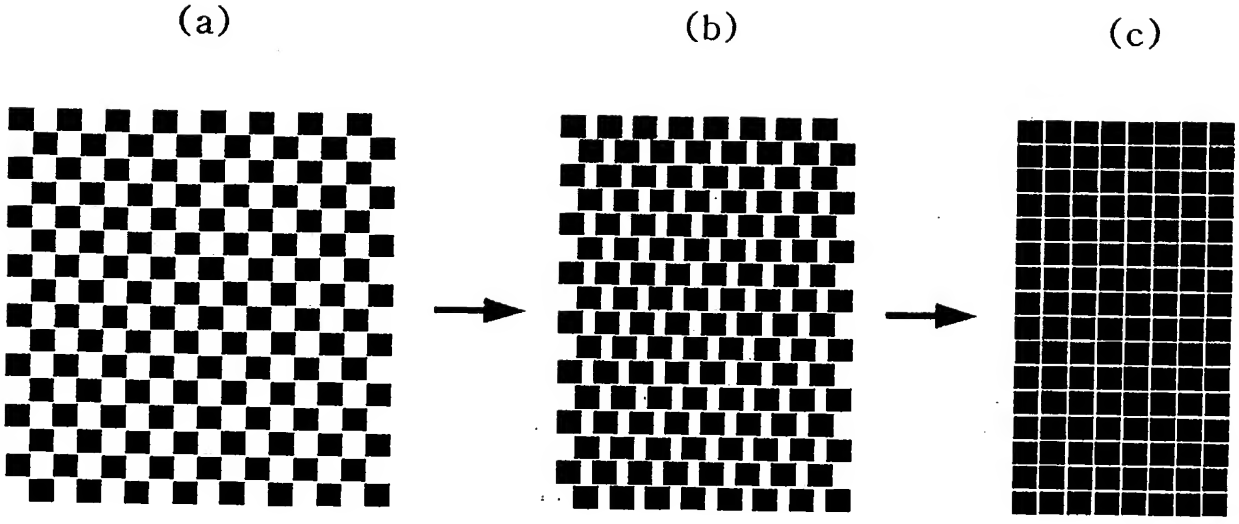
도면16


블록 매칭 처리
픽셀 위치 초기 설정
S41
AD 초기 설정
S42
S43
현 프레임의 블록과
참조 프레임의 블록간의
잔여분 AD 결정
S44
AD가 ADmin보다
큰가?
AD > ADmin
AD≦ADmin
아니오
모든 픽셀이
산출되었는가?
S45
예
복귀

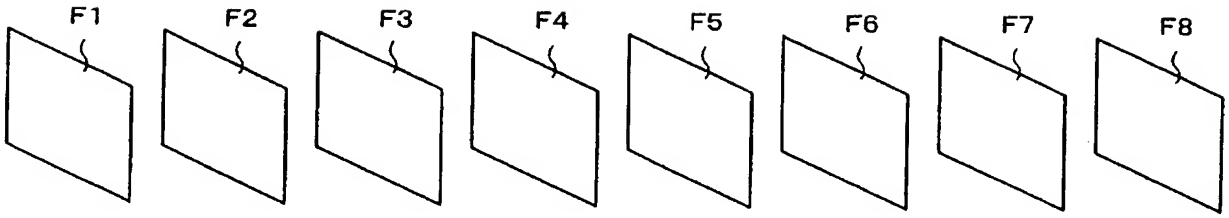
도면17



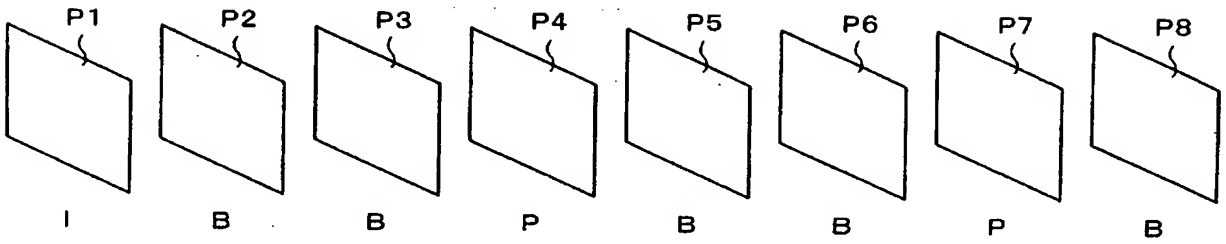
도면18



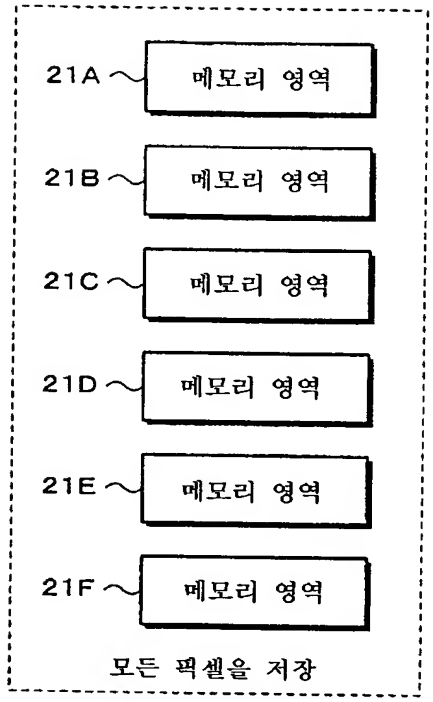
도면19a

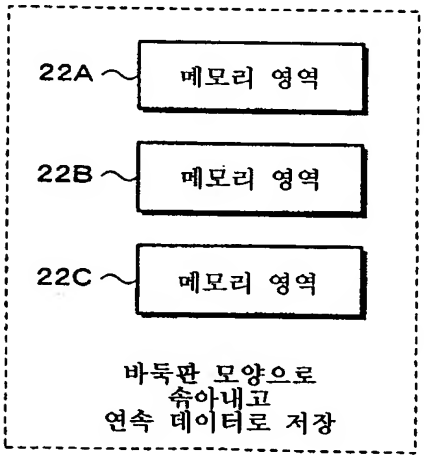


도면19b



도면20

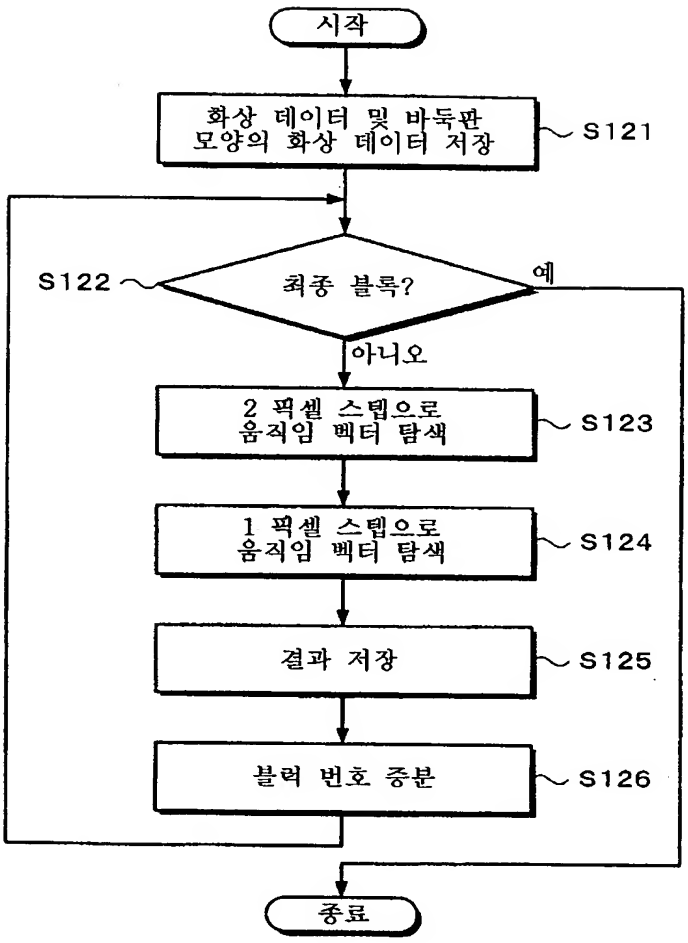

21A
메모리 영역
21B
메모리 영역
21C
메모리 영역
21D
메모리 영역
21E
메모리 영역
21F
메모리 영역
모든 픽셀을 저장


22A
메모리 영역
22B
메모리 영역
22C
메모리 영역
바둑판 모양으로
솎아내고
연속 데이터로 저장

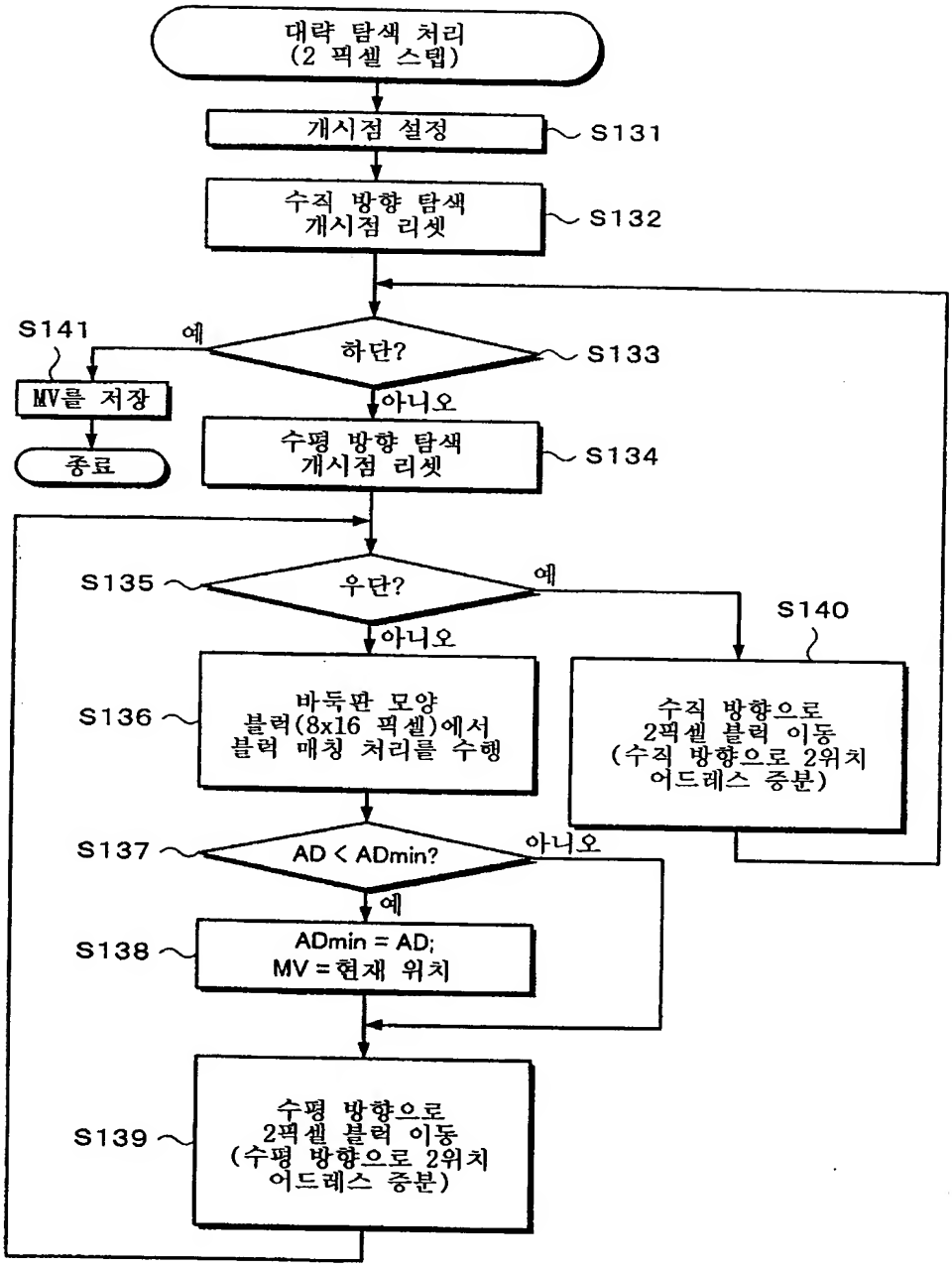
도면21

시점	T1	T2	T3	T4	T5	T6	T7	T8	T9
메모리 영역 21A	F1	F1	F1	F1	F1	F1	F7	F7	F7
메모리 영역 21B		F2	F2	F2	F2	F2	F2	F8	F8
메모리 영역 21C			F3	F3	F3	F3	F3	F3	F9
메모리 영역 21D				F4	F4	F4	F4	F4	F4
메모리 영역 21E					F5	F5	F5	F5	F5
메모리 영역 21F						F6	F6	F6	F6
메모리 영역 22A				f1	f1	f1	f7	f7	f7
메모리 영역 22B				f4	f4	f4	f4	f4	f4
메모리 영역 22C					f2	f3	f3	f5	f6
동작	엔코드 P1			엔코드 P4	엔코드 P2	엔코드 P3	엔코드 P7	엔코드 P5	엔코드 P6

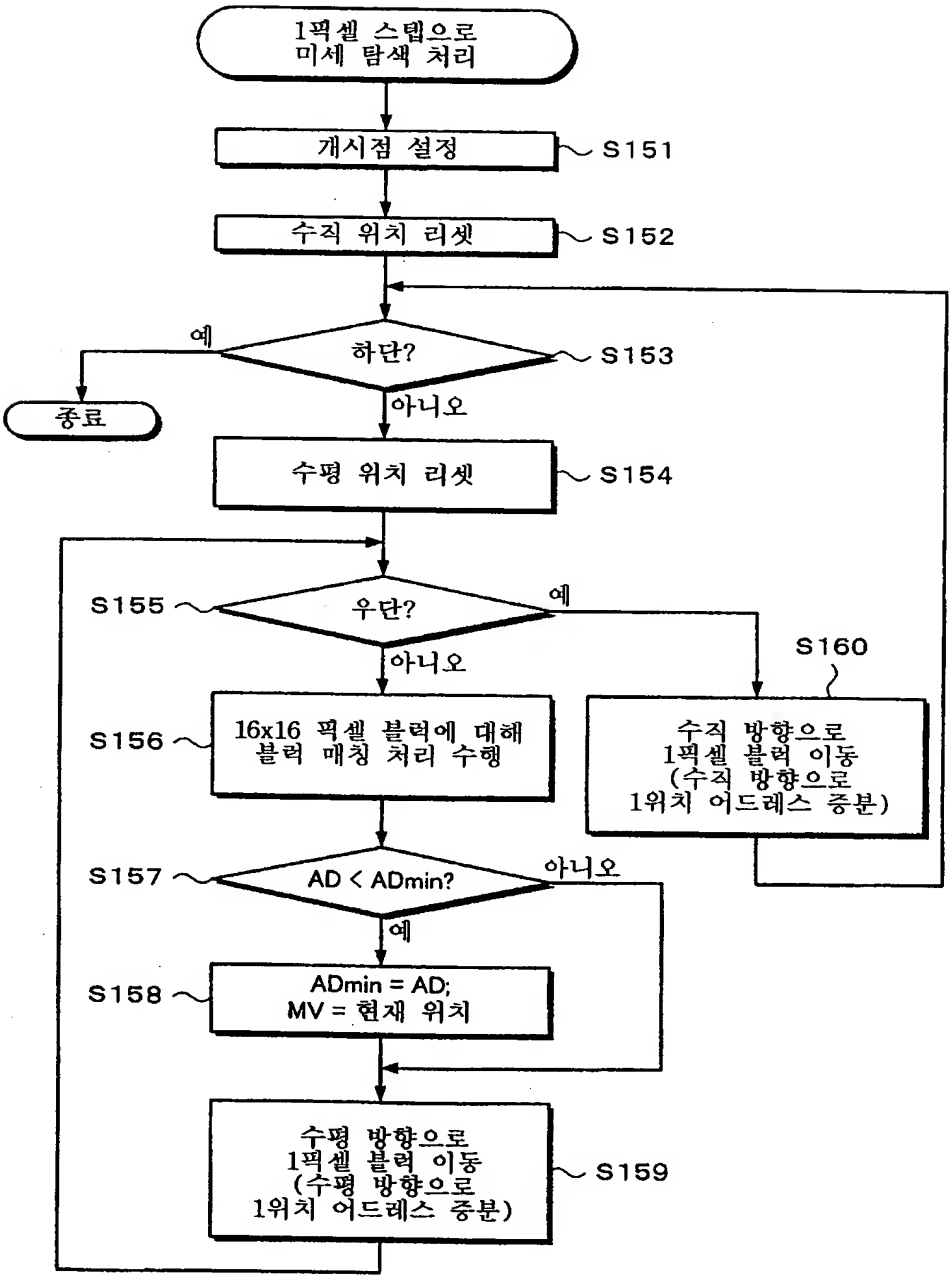
도면22


시작
화상 데이터 및 바둑판 모양의 화상 데이터 저장
S121
S122
최종 블록?
예
아니오
2 픽셀 스텝으로 움직임 벡터 탐색
S123
1 픽셀 스텝으로 움직임 벡터 탐색
S124
결과 저장
S125
블럭 번호 증분
S126
종료

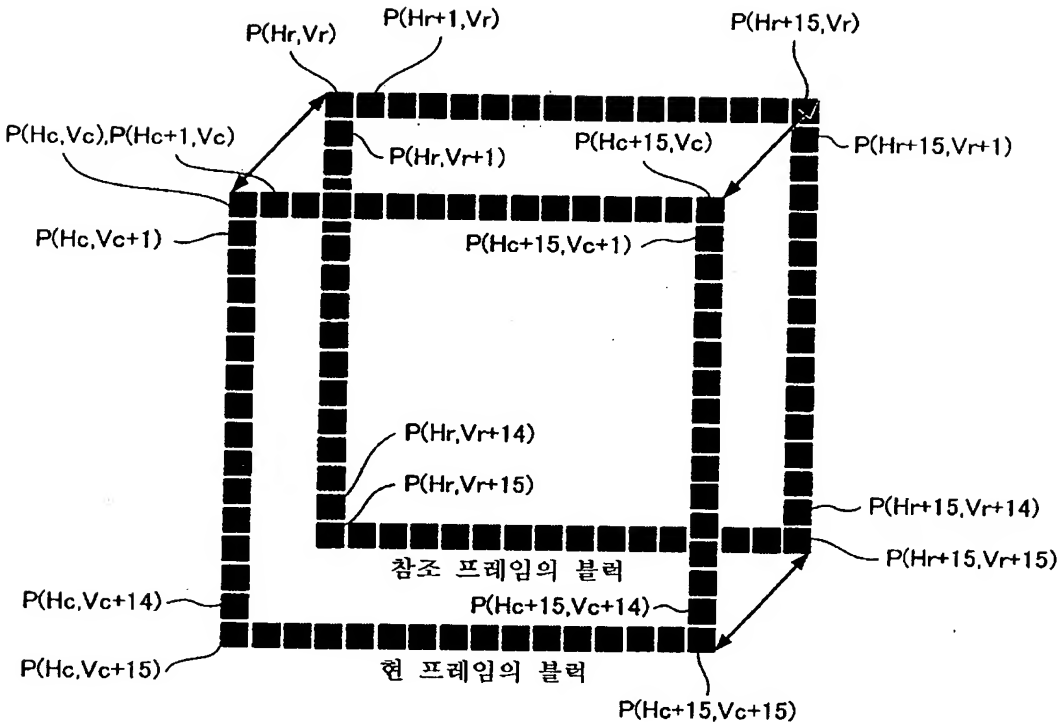
도면23


대략 탐색 처리
(2 픽셀 스텝)
개시점 설정
S131
수직 방향 탐색
개시점 리셋
S132
S141
예
하단?
S133
MV를 저장
아니오
종료
수평 방향 탐색
개시점 리셋
S134
S135
우단?
예
S140
아니오
S136
바둑판 모양
블럭(8x16 픽셀)에서
블럭 매칭 처리를 수행
수직 방향으로
2픽셀 블럭 이동
(수직 방향으로 2위치
어드레스 증분)
S137
AD < ADmin?
아니오
예
S138
ADmin = AD;
MV = 현재 위치
S139
수평 방향으로
2픽셀 블럭 이동
(수평 방향으로 2위치
어드레스 증분)

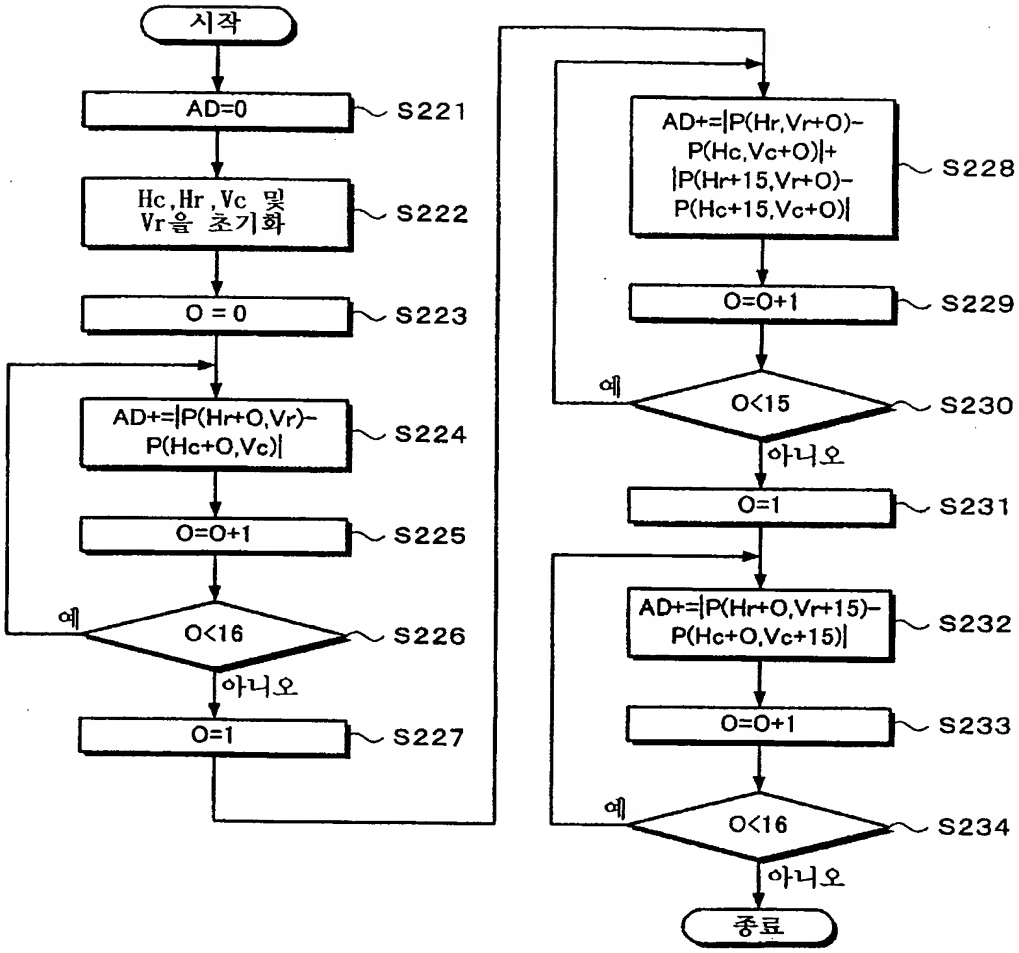
도면24


1픽셀 스텝으로
미세 탐색 처리
개시점 설정
S151
수직 위치 리셋
S152
예
하단?
S153
종료
아니오
수평 위치 리셋
S154
S155
우단?
예
아니오
S160
S156
16x16 픽셀 블럭에 대해
블럭 매칭 처리 수행
수직 방향으로
1픽셀 블럭 이동
(수직 방향으로
1위치 어드레스 증분)
S157
AD < ADmin?
아니오
예
S158
ADmin = AD;
MV = 현재 위치
수평 방향으로
1픽셀 블럭 이동
(수평 방향으로
1위치 어드레스 증분)
S159

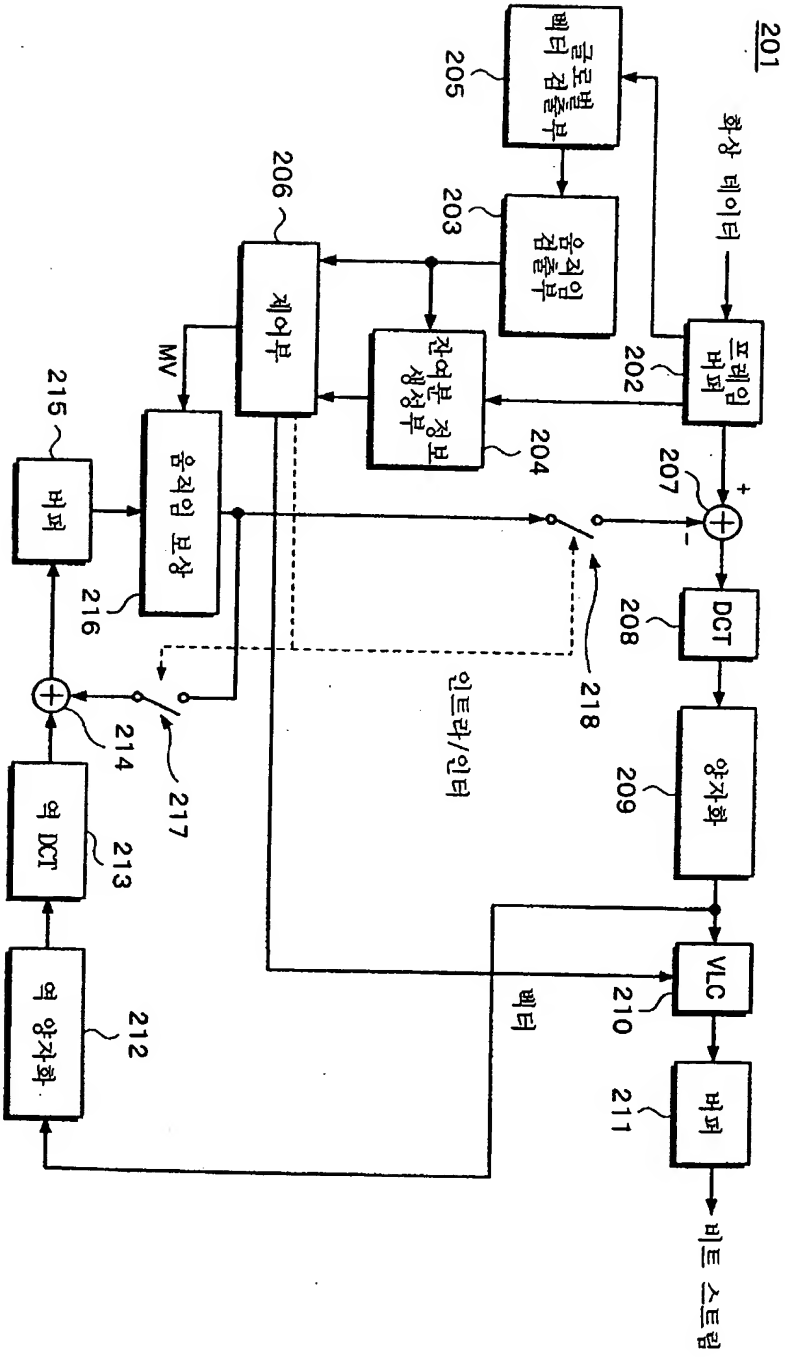
도면25


P(Hr,Vr)
P(Hr+1,Vr)
P(Hr+15,Vr)
P(Hc,Vc),P(Hc+1,Vc)
P(Hr,Vr+1)
P(Hc+15,Vc)
P(Hr+15,Vr+1)
P(Hc,Vc+1)
P(Hc+15,Vc+1)
P(Hr,Vr+14)
P(Hr,Vr+15)
P(Hr+15,Vr+14)
P(Hr+15,Vr+15)
참조 프레임의 블럭
P(Hc,Vc+14)
P(Hc+15,Vc+14)
P(Hc,Vc+15)
현 프레임의 블럭
P(Hc+15,Vc+15)

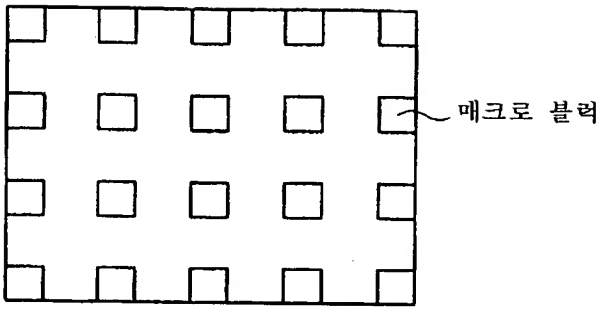
도면26


시작
AD=0
S221
Hc,Hr,Vc 및 Vr을 초기화
S222
O = 0
S223
AD+=|P(Hr+O,Vr)-P(Hc+O,Vc)|
S224
O=O+1
S225
예
O<16
S226
아니오
O=1
S227
AD+=|P(Hr,Vr+O)-P(Hc,Vc+O)|+|P(Hr+15,Vr+O)-P(Hc+15,Vc+O)|
S228
O=O+1
S229
예
O<15
S230
아니오
O=1
S231
AD+=|P(Hr+O,Vr+15)-P(Hc+O,Vc+15)|
S232
O=O+1
S233
예
O<16
S234
아니오
종료

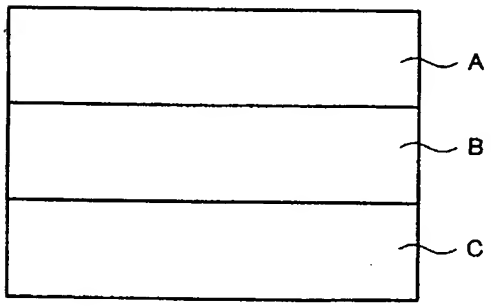
도면27


201
화상 데이터
205
글로벌 벡터 검출부
206
203
움직임 검출부
제어부
MV
215
버퍼
움직임 보상
216
202
프레임 버퍼
207
+
-
208
DCT
209
양자화
210
VLC
211
버퍼
비트 스트림
204
잔여분 정보 생성부
218
인트라/인터
214
217
213
역 DCT
212
역 양자화
벡터

도면28



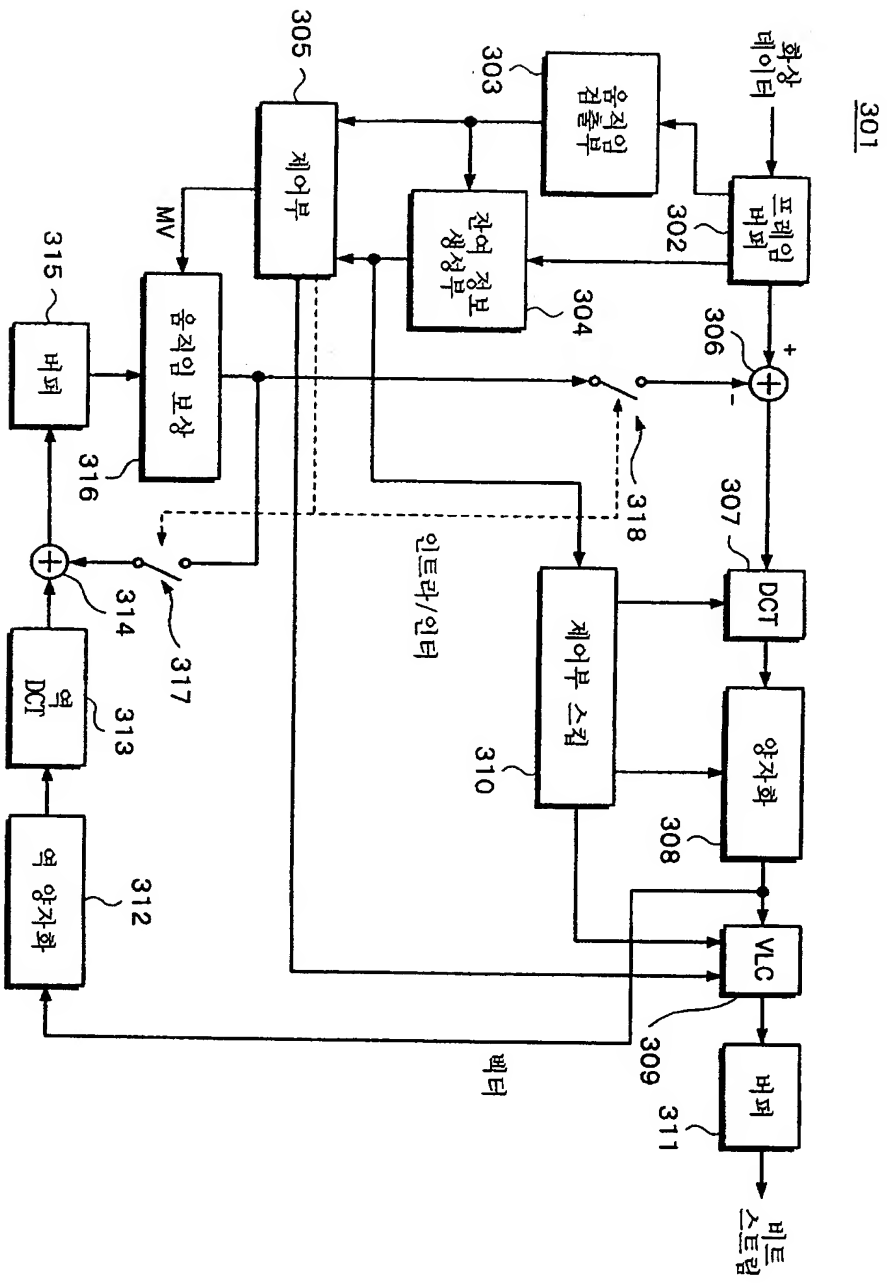
도면29



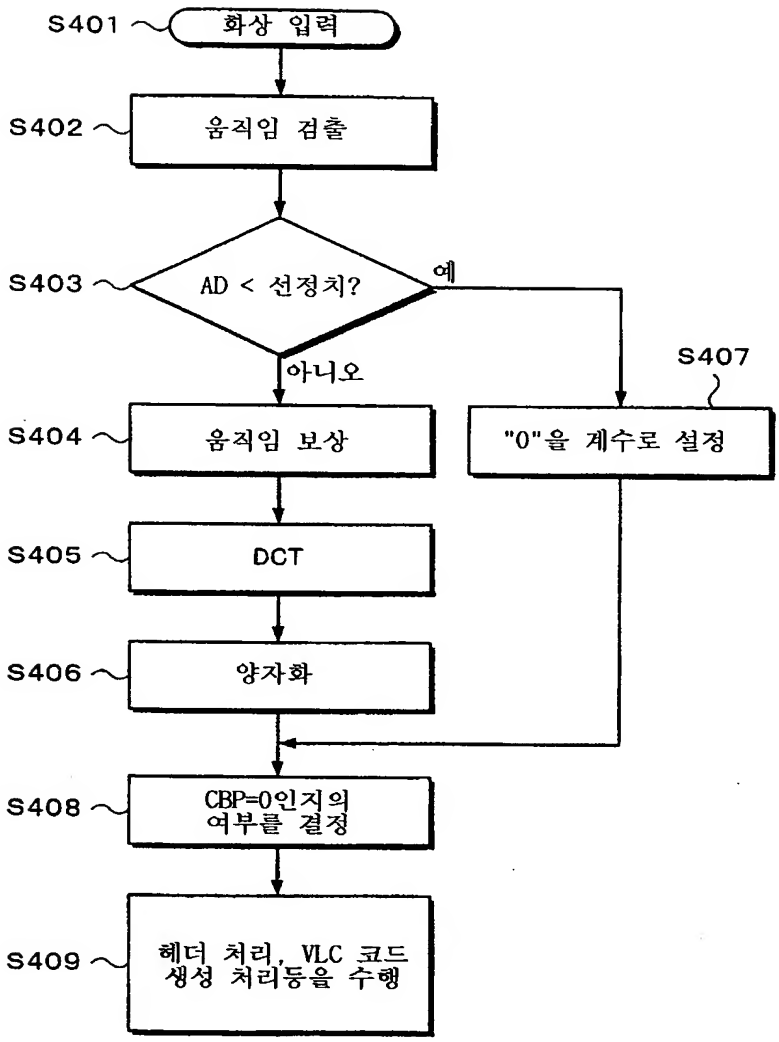
도면30

화상 입력
S301
글로벌 벡터 생성
움직임 벡터 생성
추출 처리 수행
S302
프리-스테이지 내의 움직임 검출
S303
글로벌 벡터 생성
S304
글로벌 벡터
움직임 검출
S305
각각의 매크로 블럭의 움직임 벡터 생성
S306

도면31


301
화상 데이터
프레임 버퍼
302
움직임 검출부
303
305
제어부
MV
잔여 정보 생성부
304
306
+
-
움직임 보상
316
315
버퍼
318
307
DCT
양자화
308
VLC
309
버퍼
311
비트 스트림
제어부 스킵
310
인트라/인터
314
317
역 DCT
313
역 양자화
312
베터

도면32


S401
화상 입력
S402
움직임 검출
S403
AD < 선정치?
예
아니오
S404
움직임 보상
S405
DCT
S406
양자화
S407
"0"을 계수로 설정
S408
CBP=0인지의
여부를 결정
S409
헤더 처리, VLC 코드
생성 처리등을 수행